Bounded Suboptimal Search: A Direct Approach Using Inadmissible Estimates

Jordan T. Thayer and Wheeler Ruml

University of New Hampshire

jtd7, ruml at cs.unh.edu

With Thanks To NSF-IIS0812141 and DARPA CSSG N10AP20029
sacrificing optimality can speed search

solutions could be arbitrarily bad
given a suboptimality bound $w$, find a solution with cost within a factor $w$ of optimal as quickly as possible.
Three Useful Ideas

- finding solutions and proving bounds are separate tasks
- inadmissible cost estimates can be more informed
- searching on distance is faster than cost
Three Useful Ideas

- finding solutions and proving bounds are separate tasks
- inadmissible cost estimates can be more informed
- searching on distance is faster than cost

- $A^*_\epsilon$
  Pearl and Kim, 1982
- Optimistic Search
  Thayer and Ruml, 2008
- Skeptical Search
  Thayer and Ruml, 2011
Three Useful Ideas

- finding solutions and proving bounds are separate tasks
- inadmissible cost estimates can be more informed
- searching on distance is faster than cost

Outline
- Greedy Search
- Bounded Search
- Three Ideas

EES

Conclusion
Three Useful Ideas

- finding solutions and proving bounds are separate tasks
- inadmissible cost estimates can be more informed
- searching on distance is faster than cost
Three Useful Ideas

- finding solutions and proving bounds are separate tasks
- inadmissible cost estimates can be more informed
- searching on distance is faster than cost

Explicit Estimation Search (EES) combines these three ideas.
Explicit Estimation Search
minimize solving time subject to suboptimality bound $w$
minimize solving time subject to suboptimality bound $w$

weighted A* ($f'(n) = g(n) + w \cdot h(n)$) is simple but ad hoc
(Pohl, AIJ vol 1, 1970)
minimize solving time subject to suboptimality bound \( w \)

weighted A* \( f'(n) = g(n) + w \cdot h(n) \) is simple but ad hoc

(Pohl, AIJ vol 1, 1970)

expand the node closest to a solution within the bound

\( \text{best}_d \): node estimated within bound closest to a goal
1. $h$: an admissible estimate of cost-to-go

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

finding solutions and proving bounds are separate tasks
Three Heuristic Sources Of Information

1. \( h \): an admissible estimate of cost-to-go
\[
f(n) = g(n) + h(n)
\]
finding solutions and proving bounds are separate tasks

2. \( \hat{h} \): a potentially inadmissible estimate of cost-to-go
inadmissible cost estimates can be more informed
\[
\hat{f}(n) = g(n) + \hat{h}(n)
\]
(Thayer and Rumel, ICAPS-11)
Three Heuristic Sources Of Information

Outline

- Direct Approach
- Three Heuristics
- EES

1. \( h \): an admissible estimate of cost-to-go
   \[
   f(n) = g(n) + h(n)
   \]
   finding solutions and proving bounds are separate tasks

2. \( \hat{h} \): a potentially inadmissible estimate of cost-to-go
   inadmissible cost estimates can be more informed
   \[
   \hat{f}(n) = g(n) + \hat{h}(n)
   \]
   (Thayer and Ruml, ICAPS-11)

3. \( \hat{d} \): a potentially inadmissible estimate of distance-to-go
   searching on distance is faster than cost
   (Pearl and Kim, IEEE PAMI 1982, Thayer et al, ICAPS-09)
Finding $\text{best}_d$

**Outline**

- Direct Approach
- Three Heuristics

**EES**

- Direct Approach
- Three Heuristics

**EES**

**Conclusion**

$\text{best}_f$: open node with minimum $f$

$$\arg\min_{n \in \text{open}} f(n)$$
Finding $\text{best}_f$

$best_f$: open node with minimum $f$

$$\arg\min_{n \in \text{open}} f(n)$$

$best_{\hat{f}}$: open node with minimum $\hat{f}$

$$\arg\min_{n \in \text{open}} \hat{f}(n)$$
Finding $\text{best}_{\hat{d}}$

$\text{best}_f$: open node with minimum $f$
$$\arg\min_{n\in\text{open}} f(n)$$

$\text{best}_{\hat{f}}$: open node with minimum $\hat{f}$
$$\arg\min_{n\in\text{open}} \hat{f}(n)$$

pursuing the shortest solution within the bound should be fast

$\text{best}_{\hat{d}}$: estimated $w$-admissible node with minimum $\hat{d}$
$$\arg\min_{n\in\text{open} \land \hat{f}(n) \leq w \cdot \hat{f}(\text{best}_{\hat{f}})} \hat{d}(n)$$
best_f: open node with minimum f

best_f: open node with minimum \( \hat{f} \)

best_d: estimated \( w \)-admissible node with minimum \( \hat{d} \)

node to expand next:
1. pursue the shortest solution that is within the bound.
2.
3.

in other words:
1. \( best_d \)
2.
3.
**best}_f$: open node with minimum $f$

**best}_\hat{f}$: open node with minimum $\hat{f}$

**best}_\hat{d}$: estimated $w$-admissible node with minimum $\hat{d}$

node to expand next:

1. pursue the shortest solution that is within the bound.
2. 
3. 

in other words:

1. if $\hat{f}(best}_\hat{d} \leq w \cdot f(best}_f$ then $best}_\hat{d}$
2. 
3. 

note that $f(best}_f \leq f(opt)$ and $f(n) \leq \hat{f}(n)$
best_f: open node with minimum f
best_\hat{f}: open node with minimum \hat{f}
best_\hat{d}: estimated w-admissible node with minimum \hat{d}

node to expand next:
1. pursue the shortest solution that is within the bound.
2. pursue the optimal solution.
3. in other words:
   1. if \hat{f}(best_\hat{d}) \leq w \cdot f(best_f) then best_\hat{d}
   2. else if \hat{f}(best_\hat{f}) \leq w \cdot f(best_f) then best_\hat{f}
   3.
best\_f: open node with minimum \( f \)

\( \text{best}\_\hat{f}: \) open node with minimum \( \hat{f} \)

\( \text{best}\_\hat{d}: \) estimated \( w \)-admissible node with minimum \( \hat{d} \)

node to expand next:

1. pursue the shortest solution that is within the bound.
2. pursue the optimal solution.
3. raise the lower bound on optimal solution cost.

in other words:

1. \textbf{if} \( \hat{f}(\text{best}\_\hat{d}) \leq w \cdot f(\text{best}\_f) \) \textbf{then} \( \text{best}\_\hat{d} \)
2. \textbf{else if} \( \hat{f}(\text{best}\_\hat{f}) \leq w \cdot f(\text{best}\_f) \) \textbf{then} \( \text{best}\_\hat{f} \)
3. \textbf{else} \( \text{best}\_f \)

see paper for further justification
EES Results

Outline
- Direct Approach
- Three Heuristics
- EES

Conclusion

Dock Robot

![Graph showing performance metrics for different heuristics and approaches.](Image)

- A* eps
- wA*
- Optimistic
- Skeptical
- EES
- EES Opt.
EES Results

Outline

EES
- Direct Approach
- Three Heuristics
- EES

Conclusion

Heavy Vacuum World

Suboptimality

$\log_{10}$ total raw cpu time

Suboptimality

- $wA^*$
- Optimistic
- Skeptical
- $A^*$ eps
- EES
- EES Opt.

Jordan T. Thayer (UNH)
Explicit Estimation Search (EES)

- follows directly from the objectives of bounded suboptimal search
- state of the art search bounded suboptimal search
- use inadmissible heuristics without losing bounds
- robust, works best in domains with action costs
tell your students to apply to grad school in cs at UNH!

- friendly faculty
- funding
- individual attention
- beautiful campus
- low cost of living
- easy access to Boston, White Mountains
- strong in AI, infoviz, networking, systems, bioinformatics
\[
\begin{align*}
\text{best}_f & = \arg\min_{n \in \text{open}} \hat{f}(n) \\
\text{best}_{\hat{d}} & = \arg\min_{n \in \text{open} \land \hat{f}(n) \leq w \cdot \hat{f}(\text{best}_f)} \hat{d}(n) \\
\text{best}_f & = \arg\min_{n \in \text{open}} f(n)
\end{align*}
\]
• \( p \) is the deepest node on an optimal path to \( \text{opt} \).

• \( \text{best}_f \) is the node with the smallest \( f \) value.

\[
 f(p) \leq f(\text{opt})
\]

\[
 f(\text{best}_f) \leq f(p)
\]

\( \text{best}_f \) provides a lower bound on solution cost.

determine \( \text{best}_f \) by priority queue sorted on \( f \)
Why Doesn’t $A^*_\epsilon$ Work Well?

Outline
- EES
- Conclusion
- Backup Slides
  - EES Nodes
  - Bound
    - Overhead
    - $A^*$
    - $A^*_\epsilon$ Failure
EES Overhead

Life Four-way Grid World

Suboptimality

log10 total raw cpu time

log10 total nodes generated

Skeptical
EES
wA*

EES Nodes
Bound
Overhead
A*
A* Failure

Life Four-way Grid World

Skeptical
wA*
EES
intuition: of all solutions within the bound, the nearest should be the fastest to find.

\[ f(n) = g(n) + h(n) \]

\(best_f\): generated but unexpanded node with minimum \(f\)

best-first search on two lists:

- \(open\): all generated but unexpanded nodes, sorted on \(f(n)\)
- \(focal\): all nodes where \(f(n) \leq w \cdot f(best_f)\) sorted on \(\hat{d}(n)\)

expand the best node from \(focal\)
A* Doesn’t Work Very Well

Outline
- EES
- Conclusion
- Backup Slides
  - EES Nodes
  - Bound
  - Overhead
  - $A_\epsilon^*$
  - $A_\epsilon^*$ Failure

Life Four-way Grid World

Suboptimality

total raw cpu time relative to A*

A* eps

Suboptimality

Jordan T. Thayer (UNH)
open: all generated but unexpanded nodes, sorted on $f(n)$

focal: all nodes where $f(n) \leq w \cdot f(best_f)$ sorted on $\hat{d}(n)$
**Why Doesn’t A*_ε Work Well?**

- **open**: all generated but unexpanded nodes, sorted on \( f(n) \)
- **focal**: all nodes where \( f(n) \leq w \cdot f(best_f) \) sorted on \( d(n) \)

---

**Outline**

- EES
- Conclusion

**Backup Slides**

- EES Nodes
- Bound
- Overhead
- A*_ε
- A*_ε Failure
**open**: all generated but unexpanded nodes, sorted on $f(n)$

**focal**: all nodes where $f(n) \leq w \cdot f(best_f)$ sorted on $\hat{d}(n)$

$f$ rises as search progresses ($h$ is admissible)

$best_{\hat{d}}$'s children won't remain on focal