Fast and Loose in Bounded Suboptimal Heuristic Search

Jordan Thayer and Wheeler Ruml

{jttd7, ruml} at cs.unh.edu

Ephrat Bitton

ebitton at berkeley.edu
Finding optimal solutions is prohibitively expensive.
Finding optimal solutions is prohibitively expensive.
- It's nice to limit suboptimality.
■ Finding optimal solutions is prohibitively expensive.
■ It’s nice to limit suboptimality.
■ Weighted A* is a popular method for doing that.
Finding optimal solutions is prohibitively expensive.
Its nice to limit suboptimality.
Weighted A* is a popular method for doing that.
This talk: two algorithms which are often better.
Background

Weighted A*

Strict Approach: Clamped Adaptive
Correct for underestimating \( h(n) \)
Bound correction to ensure \( w \)-admissibility

Loose Approach: Optimistic Search
Greedily search for a solution
Enforce suboptimality bound afterwards
Weighted $A^*$ (Pohl, 1970)

$A^*$ is a best first search ordered on $f(n) = g(n) + h(n)$
$A^*$ (Pohl, 1970)

$A^*$ is a best first search ordered on $f(n) = g(n) + h(n)$

Weighted $A^*$: $f'(n) = g(n) + w \cdot h(n)$
Weighted $A^*$ (Pohl, 1970)

$A^*$ is a best first search ordered on $f(n) = g(n) + h(n)$

Weighted $A^*$: $f'(n) = g(n) + w \cdot h(n)$

What does $w$ do?
- breaks ties on $f(n)$ in favor of high $g(n)$
- corrects for underestimating $h(n)$
- deepens search / emphasises greed
**Weighted A* Respects a Bound**

$p$ is a node in open on an optimal path to $opt$

\[
f(n) = g(n) + h(n) \\
f'(n) = g(n) + w \cdot h(n)
\]

\[
g(sol) \\
f'(sol) \leq f'(p) \\
g(p) + w \cdot h(p) \leq w \cdot (g(p) + h(p)) \\
w \cdot f(p) \leq w \cdot f(opt) \\
w \cdot g(opt)
\]

Therefore, $g(sol) \leq w \cdot g(opt)$
Weighted $A^*$ is a Popular Choice

- Weighted $A^*$
  - Pohl (1970)
- Dynamically Weighted $A^*$
  - Pohl (1973)
- $A_\epsilon$
  - Ghallab & Allard (1983)
- $A^*_\epsilon$
  - Pearl (1984)
- AlphA*
  - Reese & Frichs (unpublished)

Eight-way Grid Pathfinding (Unit cost)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Nodes generated (relative to $A^*$)</th>
<th>Sub-optimality Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>$dWA^*$</td>
<td>0.9</td>
<td>1.8</td>
</tr>
<tr>
<td>$A^*_\epsilon$</td>
<td>0.6</td>
<td>1.5</td>
</tr>
<tr>
<td>AlphA*</td>
<td>0.3</td>
<td>1.2</td>
</tr>
<tr>
<td>$wA^*$</td>
<td>0.0</td>
<td></td>
</tr>
</tbody>
</table>
Background
Weighted A*

Strict Approach: Clamped Adaptive
Correct for underestimating $h(n)$
Bound correction to ensure $w$-admissibility

Loose Approach: Optimistic Search
Greedily search for a solution
Enforce suboptimality bound afterwards
If \( h \) were perfect, solutions would be found in linear time.

How do we improve \( h(n) \)?

By correcting for the error in \( h(n) \)

We’ll ensure \( w \)-admissibility shortly.
Correcting $h(n)$ with one step error

Consider the single expansion:

![Tree diagram](image)

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

- $f(n)$ should remain constant across parent and child.
  - if $f(n) = g(n) + h^*(n)$ this would be true.
  - $g(n)$ is exact.
  - All the error in $f(n)$ comes from $h(n)$.

- $err_h = f(bc) - f(p)$

Track a running average of $err_h$.

\[
\hat{f}(n) = g(n) + \hat{h}(n) \\
\hat{h}(n) = h(n) \cdot (1 + err_h)
\]
Correcting $h(n)$ with one step error

Consider the single expansion:

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

- $f(n)$ should remain constant across parent and child.
  - if $f(n) = g(n) + h^*(n)$ this would be true.
  - $g(n)$ is exact.
  - All the error in $f(n)$ comes from $h(n)$.

- $err_h = f(bc) - f(p)$

Track a running average of $err_h$.

- $\hat{f}(n) = g(n) + \hat{h}(n)$
- $\hat{h}(n) = h(n) \cdot (1 + err_h)$

$\hat{h}(n)$ is inadmissible.

Clamping enforces $w$-admissibility.
Admissibility of Clamping: Weighted A*

\[ f(n) = g(n) + h(n) \]
\[ f'(n) = g(n) + w \cdot h(n) \]

\[ g(sol) \]
\[ f'(sol) \leq f'(p) \]
\[ g(p) + w \cdot h(p) \leq w \cdot (g(p) + h(p)) \]
\[ w \cdot f(p) \leq w \cdot f(opt) \]
\[ w \cdot g(opt) \]

Therefore, \( g(sol) \leq w \cdot g(opt) \)
Admissibility of Clamping: Clamped Adaptive

$p$ is a node in open on an optimal path to $opt$

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

$$\tilde{f}(n) = \min(\tilde{f}(n), w \cdot f(n))$$

$$g(sol) = \frac{\tilde{f}(sol)}{\tilde{f}(sol)} \leq \frac{\tilde{f}(p)}{\tilde{f}(p)} \leq w \cdot f(p)$$

$$w \cdot f(p) \leq w \cdot f(opt)$$

And $g(s) \leq w \cdot g(opt)$ is still true.
Empirical Evaluation

- Grid world path finding
  - Four-way and Eight-way Movement
  - Unit and Life Cost Models
  - 25%, 30%, 35%, 40%, 45% obstacles

- Temporal Planning
  - Blocksworld, Logistics, Rover, Satellite, Zenotravel

See the paper for details.
Performance of Clamped Adaptive

- Introduction
- Weighted $A^*$
- Clamped Adaptive
  - Improving $wA^*$
  - Correcting $h(n)$
  - $w$-Admissibility
- Performance
- Optimistic Search
- Conclusion

Four-way Grid Pathfinding (Unit cost)

Nodes generated (relative to $A^*$)

Sub-optimality Bound

- $wA^*$
- Clamped Adaptive
Performance of Clamped Adaptive

Introduction

Weighted $A^*$

Clamped Adaptive
- Improving $wA^*$
- Correcting $h(n)$
- $w$-Admissibility
- Performance

Optimistic Search

Conclusion

The diagram shows the comparison between $wA^*$ and Clamped Adaptive in terms of nodes generated relative to $A^*$ across different sub-optimality bounds. The y-axis represents the ratio of nodes generated, and the x-axis represents the sub-optimality bound.

zenotravel (problem 2)
Performance of Clamped Adaptive

Introduction

Weighted \( A^* \)

Clamped Adaptive
- Improving \( wA^* \)
- Correcting \( h(n) \)
- \( w \)-Admissibility
- Performance

Optimistic Search

Conclusion

- Nodes generated (relative to \( A^* \))
- Sub-optimality Bound
- satellite (problem 2)
- \( wA^* \) - Black
- Clamped Adaptive - Red
Performance of Clamped Adaptive

**Introduction**

**Weighted \( A^* \)**

**Clamped Adaptive**
- Improving \( wA^* \)
- Correcting \( h(n) \)
- \( w \)-Admissibility
- Performance

**Optimistic Search**

**Conclusion**

---

logistics (problem 3)

Nodes generated (relative to \( A^* \))

- Clamped Adaptive
- \( wA^* \)

---

Sub-optimality Bound

- 1
- 2
- 3
Clamped Adaptive: Summary

Clamped Adaptive:

- On-line heuristic correction seems promising
  Performance varies
  Does well for small bounds
  Fails to become greedy
- No parameter tuning needed
- Clamping for admissibility of inadmissible heuristics
Background
Weighted A*

Strict Approach: Clamped Adaptive
Correct for underestimating $h(n)$
Bound correction to ensure $w$-admissibility

Loose Approach: Optimistic Search
Greedily search for a solution
Enforce suboptimality bound afterwards
Weighted $A^*$ Respects a Bound

\[
\begin{align*}
  f(n) &= g(n) + h(n) \\
  f'(n) &= g(n) + w \cdot h(n)
\end{align*}
\]

\[
\begin{align*}
  g(sol) &\leq f'(sol) \\
  &\leq f'(p) \\
  &\leq g(p) + w \cdot h(p) \\
  &\leq w \cdot (g(p) + h(p)) \\
  w \cdot f(p) &\leq w \cdot f(opt) \\
  &\leq w \cdot g(opt)
\end{align*}
\]

Therefore, $g(sol) \leq w \cdot g(opt)$
Weighted $A^*$ Respects the Bound and Then Some

\[ f(n) = g(n) + h(n) \]
\[ f'(n) = g(n) + w \cdot h(n) \]

\[
\begin{align*}
  g(sol) & \leq f'(sol) \\
  f'(sol) & \leq f'(p) \\
  g(p) + w \cdot h(p) & \leq w \cdot (g(p) + h(p)) \\
  w \cdot f(p) & \leq w \cdot f(opt) \\
  w \cdot g(opt) & \leq w \cdot g(opt) \\
  g(p) + w \cdot h(p) & \leq w \cdot g(p) + w \cdot h(p)
\end{align*}
\]
Solution Quality v. Bound

- $wA^*$ returns solutions better than the bound.
- Be optimistic
- Run with higher weight

How do we guarantee a suboptimality bound?
\[ f(p) \leq f(\text{opt}) \]
\[ f(f_{\text{min}}) \leq f(p) \]
\[ f_{\text{min}} \text{ provides a lower bound on solution cost.} \]

Determine \( f_{\text{min}} \) by priority queue sorted on \( f \)

Optimistic Search: Run a greedy search

Expand \( f_{\text{min}} \) until \( w \cdot f_{\text{min}} \geq f(\text{sol}) \)

- \( p \) is the deepest node on an optimal path to \( \text{opt} \)
This Paper:

- Grid world path finding
  Four-way and Eight-way Movement
  Unit and Life Cost Models
  25
- Temporal Planning
  Blocksworld, Logistics, Rover, Satellite, Zenotravel

To Appear in ICAPS:

- Traveling Salesman
  Unit Square
  Pearl and Kim Hard
- Sliding Tile Puzzles
  Korf’s 100 15-puzzle instances

See papers for details.
Performance of Optimistic Search

Introduction

Weighted $A^*$

Clamped Adaptive

Optimistic Search
- Loose Bounds
- Solution Quality
- $w$-Admissibility
- Performance

Conclusion

---

Pearl and Kim Hard

Node Generations Relative to $A^*$

$wA^*$

Optimistic

Sub-optimality bound
Performance of Optimistic Search

Introduction

Weighted $A^*$

Clamped Adaptive

Optimistic Search

- Loose Bounds
- Solution Quality
- $w$-Admissibility
- Performance

Conclusion

Korf’s 15 Puzzles

Node Generations Relative to IDA*

Sub-optimality bound

$wA^*$

Optimistic
Four-way Grid Pathfinding (Unit cost)

- Loose Bounds
- Solution Quality
- $w$-Admissibility
- Performance
Conclusion

Clamped Adaptive:
- On-line heuristic correction seems promising.
- No parameter tuning needed.

Optimistic Search:
- Performance is predictable.
- Current results are good, could be improved.

We have two algorithms that can outperform weighted $A^*$

We can use arbitrary heuristics for $w$-admissible search.
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
Bounded Anytime Weighted A*

Korf’s 15 Puzzles

Graph showing node generations relative to IDA* for different algorithms.

- BAwA*
- wA*
- Optimistic

Sub-optimality bound

Node Generations Relative to IDA*

Introduction

Weighted A*

Clamped Adaptive

Optimistic Search

Conclusion

Bounded Anytime Weighted A*
Bounded Anytime Weighted $A^*$

Pearl and Kim Hard

Node Generations Relative to $A^*$

Sub-optimality bound

BAwA* — green
wA* — black
Optimistic — blue
Duplicate Dropping can be Important

Introduction
Weighted $A^*$
Clamped Adaptive
Optimistic Search
Conclusion
Duplicate Dropping

Four-way Grid Pathfinding (Unit cost)

- $wA^*$
- $wA^*$ dd

Nodes generated (relative to $A^*$)
Sub-optimality Bound
Sometimes it isn’t

Korf’s 15 puzzles

Node Generations Relative to IDA*

Sub-optimality bound

WA* dd

WA*

0.09

0.06

0.03

0.0