Tunable Suboptimal Heuristic Search

Stephen Wissow[1], Fanhao Yu[2], Wheeler Ruml[1]

[1] University of New Hampshire, USA
[2] Nashua High School South, USA

Thanks to the NSF-BSF program for support via NSF grant 2008594.

Contributions

1. new problem setting: tunable suboptimal heuristic search
   - what practitioners really want?
2. new algorithm: Speed*
   - complete
3. experimental survey of tunable setting
   - Bead usually best but fails in domains with dead-ends
   - Speed* best in domains with dead-ends
   - best bounded-suboptimal $\neq$ best tunable

Search Settings and Objectives

optimal: minimize solution cost

- A* (Hart, Nilsson, and Raphael 1968, $f = g + h$)

greedy: minimize solving time

- GBFS (Michie and Ross 1969, h)
- Speedy (Ruml and Do 2007, d)

Search Settings and Objectives

optimal: minimize solution cost

- A* (Hart, Nilsson, and Raphael 1968, $f = g + h$)

greedy: minimize solving time

- GBFS (Michie and Ross 1969, h)
- Speedy (Ruml and Do 2007, d)

bounded suboptimal:
minimize time subject to relative cost bound

- $w\mathrm{A}^*$ (Pohl 1970, $f' = g + w \cdot h$)
- RR-d (Fickert, Gu, and Ruml 2022, $d, \hat{f}, f$)

Search Settings and Objectives

optimal: minimize solution cost

- A* (Hart, Nilsson, and Raphael 1968, $f = g + h$)

greedy: minimize solving time

- GBFS (Michie and Ross 1969, h)
- Speedy (Ruml and Do 2007, d)

bounded suboptimal:
minimize time subject to relative cost bound

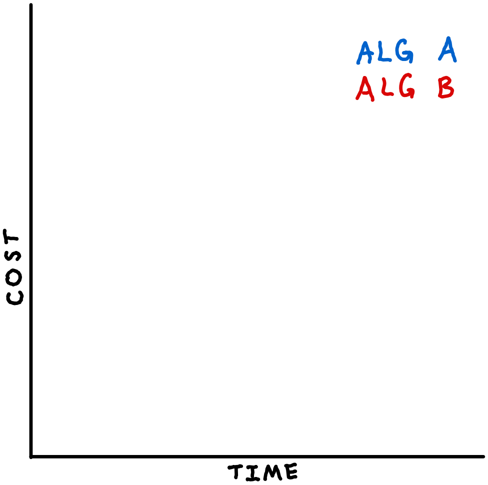
- $w\mathrm{A}^*$ (Pohl 1970, $f' = g + w \cdot h$)
- RR-d (Fickert, Gu, and Ruml 2022, $d, \hat{f}, f$)

new setting: tunable suboptimal:
best cost-time trade-off—no guarantees

Pedagogical Example

bounded suboptimal:

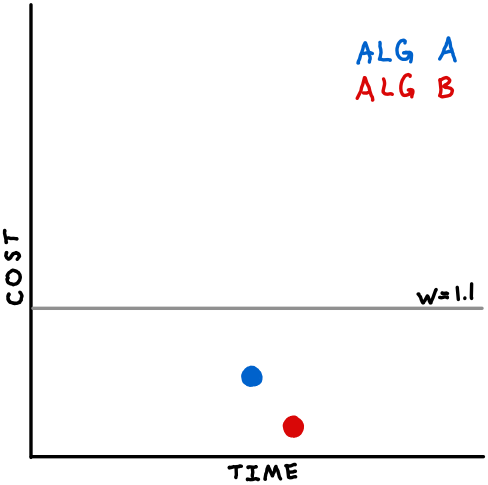
- minimize time subject to relative cost bound
- left is good



Pedagogical Example

bounded suboptimal:

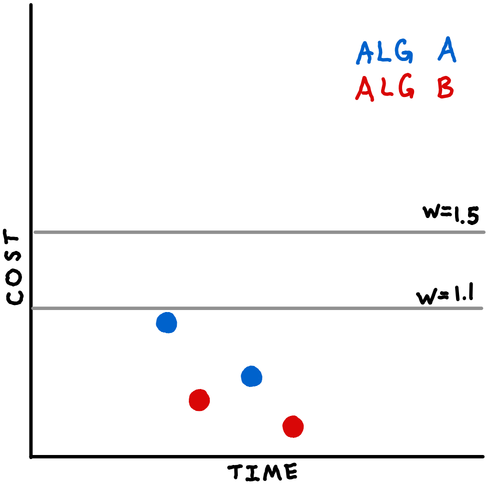
- minimize time subject to relative cost bound
- left is good



Pedagogical Example

bounded suboptimal:

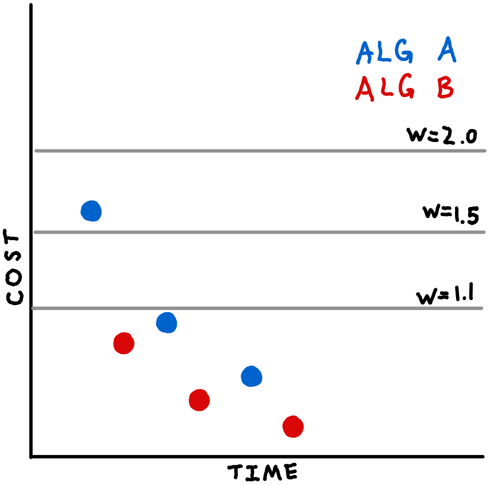
- minimize time subject to relative cost bound
- left is good



Pedagogical Example

bounded suboptimal:

- minimize time subject to relative cost bound
- left is good



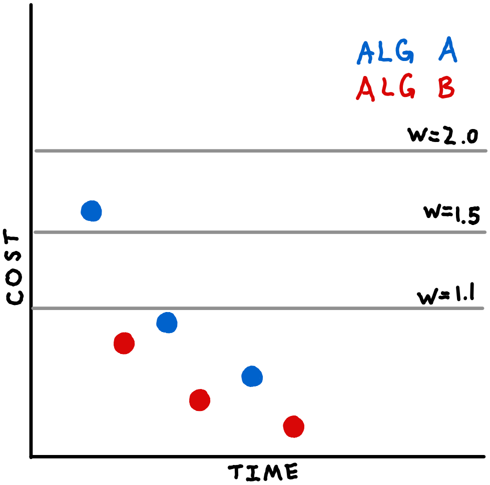
Pedagogical Example

bounded suboptimal:

- minimize time subject to relative cost bound
- left is good

tunable:

- best cost-time trade-off
- down and left is good



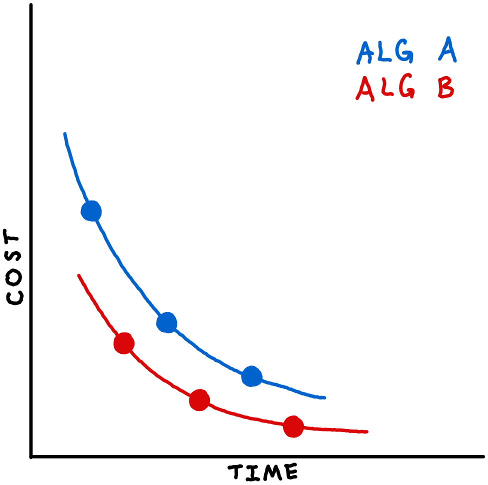
Pedagogical Example

bounded suboptimal:

- minimize time subject to relative cost bound
- left is good

tunable:

- best cost-time trade-off
- down and left is good



best bounded suboptimal $\neq$ best tunable!

new algorithm: Speed*

- best-first on $f^{\dagger}(n) = g(n) + h(n) + s' \cdot d(n)$
  - interpolates between A* and Speedy
- see paper for completeness proof, post hoc suboptimality bound

simple design expressly for tunable setting!

Contributions

1. tunable suboptimal heuristic search
2. Speed*
3. experimental survey

Algorithms Compared

1. $w\mathrm{A}^*$: simple popular bounded suboptimal
2. RR-d: state-of-the-art bounded suboptimal
3. Bead: incomplete but fast and tunable
4. Rectangle: state-of-the-art anytime, terminated at solution k
5. Speed*: explicitly tunable

Reference algorithms:

6. GBFS: limit of $w\mathrm{A}^*$ (big w)
7. Speedy: limit of Speed* (big s), RR-d (big w)
8. Hill-Climbing: limit of Beam search ($b = 1$)

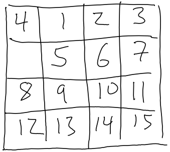
Problem Domains

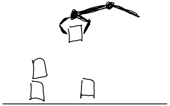
no dead-ends:

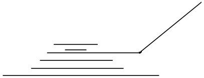
1. sliding tiles (unit and non-unit cost)
2. blocks world
3. pancake (unit and non-unit cost)

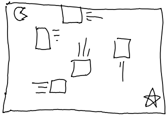
dead-ends:

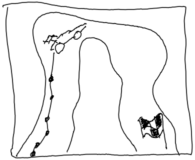
4. traffic
5. New Hampshire racetrack (crash is dead-end)

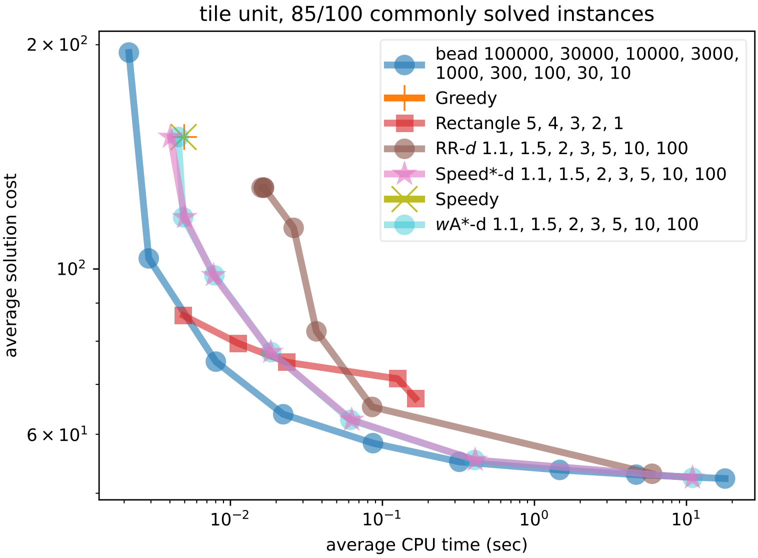




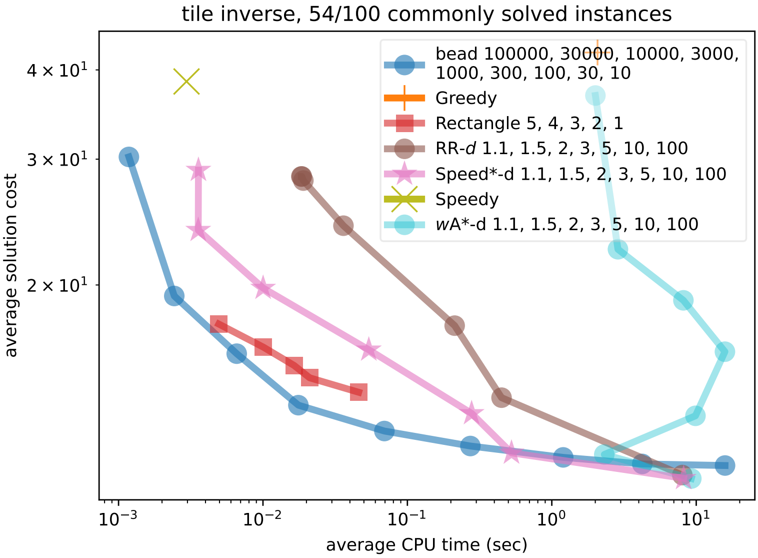




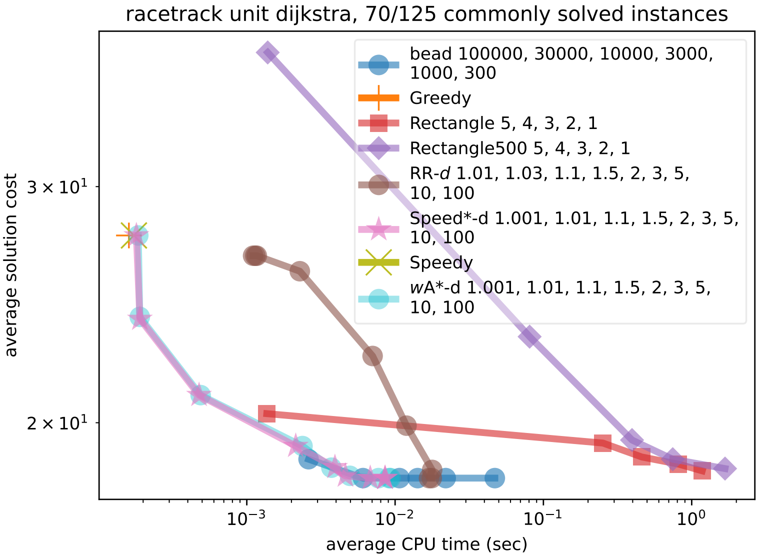




Bead has best cost-time trade-off—Speed* outperforms RR-d



Speed* outperforms wA^* on non-unit cost



Bead fails on dead-ends

Speed* robust to dead-ends

Results: Summary

- Bead awesome on domains without dead-ends
- Speed* robust to dead-ends
- Speed* outperforms $w\mathrm{A}^*$ and RR-d

Contributions

1. new problem setting: tunable suboptimal heuristic search
   - no guarantees, but what practitioners really care about?
2. new algorithm: Speed*
   - complete
3. experimental survey of tunable setting
   - Bead usually best but fails in domains with dead-ends (incompleteness)
   - Speed* best in domains with dead-ends
   - best bounded-suboptimal $\neq$ best tunable

s' computation for Speed*

- help account for inter-domain variation in relative magnitudes of d and h
- make Speed* behave like $w\mathrm{A}^*$ for given value of s or w on unit cost
- computed once at start and held constant throughout search

$$f^\dagger(n) = g(n) + h(n) + s' \cdot d(n)$$

$$s' = (s-1) \cdot \frac{h(n_i)}{d(n_i)}$$

$$s \in [1, \inf)$$

initial state n_i

Speed*'s Post Hoc Suboptimality Bound

for solution cost C, min f value on frontier at termination, optimal cost C^*:

$$b := \frac{C}{f_{\min}} \geq \frac{C}{C^*}$$

borrows approach from anytime setting; can apply to Speed* too

Why is searching on d faster than h?

hypothesis: number and size of local minima affect performance

- Pearl and Kim (1982)
- Thayer and Ruml (2009)
- Wilt and Ruml (2014)
- Cohen and Beck (2018)

many questions remain in use of distance in suboptimal search

Limitations

- other settings convertible to tunable (like anytime/Rectangle):
  - real-time search: return next action within absolute time bound
- really hard problems
  - imposed 1 min solving time limit in experiments
  - e.g. 81-puzzle (sliding tiles)

Real-Time Search

- interpolates between A* and Hill-Climbing
- in each iteration: absolute time bound on selecting next action to commit to
  - tunable 'knob'
- terminate at first solution
- Hill-Climbing is Real-Time with time bound of 0
- time bound similar to beam width but more robust
- notoriously subject to dead-ends like Beam search
- not commonly used in tunable setting

Extending Previous Work

tunable setting implied in previous work:

- Wilt, Thayer, and Ruml (2010)
- Wilt, Thayer, and Ruml (2011)

present work (Wissow, Yu, and Ruml 2024) is first tunable survey to compare state-of-the-art RR-d and Rectangle.

Bounded Cost and Contract Search Settings

- bounded cost search: guarantee solution cost within absolute bound c of optimal: $C \leq c + C^*$
  - understanding effect of c on search behavior requires familiarity with range of typical solution costs, including C^*
  - units of absolute bound c vary across problem domains, impeding interpretability
  - e.g. Potential Search (Stern, Puzis, and Felner 2011)
- contract search: minimize solution cost subject to absolute bound on search time
  - requires reasoning about two uncertain values:
    - how likely to find cheapest solution in subtree of given node
    - how likely to find (any) solution in subtree of given node *before deadline*
  - sparse previous work

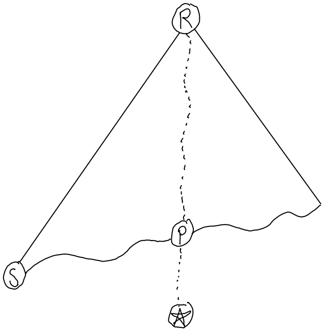
wA*'s Pathological Behavior

Wilt and Ruml (2012):

- correlation between $d^*(n)$ and $h(n)$: varies by domain
- domains with low correlation $\rightarrow$ GBFS can fare poorly
- not $h^*(n)$–$h(n)$ correlation
- not % error $\frac{h^*(n)-h(n)}{h^*(n)}$
- not local minima size

$w\mathrm{A}^*$ Bounded Suboptimality Proof Sketch

for returned solution s, and unexpanded p along an optimal solution path



$$
\begin{aligned}
f'(s) &\leq f'(p) && s \text{ expanded before } p \\
g(s) &\leq g(p) + w \cdot h(p) && \text{definition of } f'\text{, goal-aware } h \\
&\leq w \cdot g(p) + w \cdot h(p) && \text{algebra} \\
&= w \cdot f(p) && \text{definition of } f \\
&\leq w \cdot C^* && \text{admissible } h
\end{aligned}
$$

Problem Domain Citations

1. sliding tile puzzle: Korf (1985, '15-puzzle')
2. blocks world: Slaney and Thiébaux (2001)
3. pancake: Helmert (2010), non-unit: Hatem and Ruml (2014)

dead-ends:

4. traffic: Kiesel, Burns, and Ruml (2015)
5. New Hampshire racetrack: variant of Gardner (1973) that adds dead-ends

References I

Cohen, E.; and Beck, J. C. 2018. Local Minima, Heavy Tails, and Search Effort for GBFS. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18)*, 4708–4714.

Fickert, M.; Gu, T.; and Ruml, W. 2022. New Results in Bounded-Suboptimal Search. In *Proceedings of the Thirty-sixth AAAI Conference on Artificial Intelligence (AAAI-22)*.

Gardner, M. 1973. Mathematical Games. *Scientific American*, 228(5): 102–107.

Hart, P. E.; Nilsson, N. J.; and Raphael, B. 1968. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions of Systems Science and Cybernetics*, SSC-4(2): 100–107.

Hatem, M.; and Ruml, W. 2014. Bounded Suboptimal Search in Linear Space: New Results. In *Proceedings of the Seventh Annual Symposium on Combinatorial Search (SoCS-14)*.

References II

Helmert, M. 2010. Landmark Heuristics for the Pancake Problem. In Felner, A.; and Sturtevant, N. R., eds., *Proceedings of the Third Annual Symposium on Combinatorial Search, SOCS 2010, Stone Mountain, Atlanta, Georgia, USA, July 8-10, 2010*, 109–110. AAAI Press.

Kiesel, S.; Burns, E.; and Ruml, W. 2015. Achieving goals quickly using real-time search: experimental results in video games. *Journal of Artificial Intelligence Research*, 54: 123–158.

Korf, R. E. 1985. Iterative-Deepening-A*: An Optimal Admissible Tree Search. In *Proceedings of IJCAI-85*, 1034–1036.

Michie, D.; and Ross, R. 1969. Experiments with the Adaptive Graph Traverser. In *Machine Intelligence 5*, 301–318.

Pearl, J.; and Kim, J. H. 1982. Studies in Semi-Admissible Heuristics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-4(4): 391–399.

References III

Pohl, I. 1970. Heuristic Search Viewed as Path Finding in a Graph. *Artificial Intelligence*, 1: 193–204.

Ruml, W.; and Do, M. B. 2007. Best-first Utility-guided Search. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-07)*, 2378–2384.

Slaney, J. K.; and Thiébaux, S. 2001. Blocks World revisited. *Artif. Intell.*, 125(1-2): 119–153.

Stern, R.; Puzis, R.; and Felner, A. 2011. Potential Search: A Bounded-Cost Search Algorithm. In *Proceedings of the Twenty-first International Conference on Automated Planning and Scheduling (ICAPS-11)*.

Thayer, J. T.; and Ruml, W. 2009. Using Distance Estimates in Heuristic Search. In *Proceedings of the Nineteenth International Conference on Automated Planning and Scheduling (ICAPS-09)*.

References IV

Wilt, C.; and Ruml, W. 2012. When Does Weighted A* Fail? In *Proceedings of SoCS*.

Wilt, C.; and Ruml, W. 2014. Speedy Versus Greedy Search. In *Proceedings of the Seventh International Symposium on Combinatorial Search (SoCS-14)*.

Wilt, C.; Thayer, J.; and Ruml, W. 2010. A Comparison of Greedy Search Algorithms. In *Proceedings of the Symposium on Combinatorial Search (SoCS-10)*.

Wilt, C.; Thayer, J.; and Ruml, W. 2011. Selecting a Greedy Search Algorithm. Technical Report TR-10-07, University of New Hampshire.

Wissow, S.; Yu, F.; and Ruml, W. 2024. Tunable Suboptimal Heuristic Search. In *Proceedings of the Seventeenth International Symposium on Combinatorial Search (SoCS-24)*.