Finding Acceptable Solutions Faster Using Inadmissible Information

Jordan Thayer and **Wheeler Ruml**
⟨ jtd7, ruml ⟩ at cs.unh.edu
Department of Computer Science
University of New Hampshire
Durham, NH 03824 USA

Abstract

Bounded suboptimal search algorithms attempt to find a solution quickly while guaranteeing that the cost does not exceed optimal by more than a desired factor. These algorithms generally use a single admissible heuristic both for guidance and guaranteeing solution quality. We present a new approach to bounded suboptimal search that separates these roles, consulting multiple sources of potentially inadmissible information to determine search order and using admissible information to guarantee quality. An empirical evaluation across six benchmark domains shows the new approach has better overall performance.

Explicit Estimation Search (EES)

The objective of bounded suboptimal search, finding a solution within the bound as quickly as possible, suggests the following search order: for all nodes that appear to be on a path to a solution within the bound, expand the node that seems closest to a goal. EES follows this principle as directly as possible while strictly guaranteeing the bound. To accomplish this EES uses $\widehat{h}$, an unbiased estimate of the cost-to-go, as opposed to h, a lower bound, and $\widehat{d}$, an estimate of the number of actions to go. EES relies on two node evaluation functions, f and $\widehat{f}$. $f(n) = g(n) + h(n)$ is the traditional cost function of A* (Hart, Nilsson, and Raphael 1968) and provides a lower bound on the cost of an optimal solution through n. $\widehat{f}(n) = g(n) + \widehat{h}(n)$ is an unbiased estimate of the cost of the best solution through n. $\widehat{h}$ and $\widehat{d}$ can be supplied by the user, they may be constructed by correcting h and d during the search (Thayer, Ruml, and Bitton 2008), or they may be constructed using offline techniques before search begins (Samadi, Felner, and Schaeffer 2008).

EES expands one of the following nodes:

$$f_{min} = \operatorname*{argmin}_{n \in open} f(n)$$

$$best_{\widehat{f}} = \operatorname*{argmin}_{n \in open} \widehat{f}(n)$$

$$best_{\widehat{d}} = \operatorname*{argmin}_{n \in open \wedge \widehat{f}(n) \leq w \cdot \widehat{f}(best_{\widehat{f}})} \widehat{d}(n)$$

f_{min} is the node with the lowest f value among all unexpanded nodes. $best_{\widehat{f}}$ is the node with the lowest predicted solution cost. $best_{\widehat{d}}$ is selected from a restricted set of nodes, those whose $\widehat{f}$ value is within a factor w of $\widehat{f}(best_{\widehat{f}})$, or more plainly, from those nodes we suspect lead to a solution within the desired bound. Of these, $best_{\widehat{d}}$ is the node nearest to a goal. EES chooses the next node to expand using the rules:

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

We first consider $best_{\widehat{d}}$, the node which appears to be closest to a goal, as dictated by the objective of bounded suboptimal search. We only return $best_{\widehat{d}}$ if the estimated cost of a solution through it is within a factor w of the lower bound on the cost of an optimal solution. If $best_{\widehat{d}}$ is unsuitable, $best_{\widehat{f}}$ is examined. Expanding it may produce a high quality solution, and expanding it instead of f_{min} avoids the thrashing behavior of A^*_ϵ first described in Thayer, Ruml, and Kreis (2009). If neither $best_{\widehat{f}}$ nor $best_{\widehat{d}}$ were within the bound, we return f_{min}, potentially raising our lower bound, allowing us to consider $best_{\widehat{d}}$ or $best_{\widehat{f}}$ next. This expansion order strictly enforces suboptimality bounds (Thayer and Ruml 2010).

Performance

To evaluate EES, we implemented and compared against all other bounded suboptimal searches we found in the literature: A_ϵ (Ghallab and Allard 1983), AlphA* (Reese 1999), revised dynamically weighted A* (Thayer and Ruml 2009), A^*_ϵ (Pearl and Kim 1982), optimistic search (Thayer and Ruml 2008), and skeptical search (Thayer and Ruml 2010). We test on vacuum world, dynamic robot navigation, the sliding tile puzzle, the traveling salesman problem, the pancake puzzle, and grid navigation problems.

Table 1 shows the relative performance of the algorithms at different suboptimality bounds across all of the domains in our evaluation. We present the CPU time consumed relative to that used by EES, averaged over all domains. These figures give us a quantitative sense of the relative performance of the algorithms. We see that, with a single excep-

Cost Bound	1.5	1.75	2.	3.	5.
optimistic	1.6	1.5	1.6	2.1	2.1
skeptical	2.6	4.7	4.9	5.1	13.9
A_ϵ^*	50.4	44.8	28.5	1.8	0.6
wA*	4.1	3.4	2.8	3.7	2.4
A_ϵ	911.4	857.7	683.2	624.9	614.3
AlphA*	126.6	140.1	181.6	282.2	315.0
Clamped	8.3	10.1	11.6	67.0	85.8
rdwA*	374.1	316.9	245.1	101.0	128.1

Table 1: CPU time relative to EES

CPU	1^{st}	2^{nd}	3^{rd}	4^{th}	$> 4^{th}$
EES	0	3	3	0	0
Optimistic	0	2	1	3	0
Skeptical	3	0	0	1	2
A_ϵ^*	2	1	1	0	3
wA*	1	0	0	2	3
A_ϵ	0	0	0	0	6
AlphA*	0	0	0	0	6
Clamped	0	0	1	0	5
rdwA*	0	0	0	0	6
Generated	1^{st}	2^{nd}	3^{rd}	4^{th}	$> 4^{th}$
EES	2	3	1	0	0
Optimistic	0	1	2	3	0
Skeptical	3	0	0	1	2
A_ϵ^*	1	1	1	0	3
wA*	1	0	0	2	3
A_ϵ	0	0	0	0	6
AlphA*	0	0	0	0	6
Clamped	0	0	1	0	5
rdwA*	0	0	0	0	6

Table 2: Rank across all benchmarks

tion of A_ϵ^* run with a bound of 5, EES consumes less time than any other approach, despite its considerable overhead.

Table 2 summarizes performance of the algorithms on a per domain basis, showing the number of times each algorithm achieved each ranking across all benchmarks when ranked by the number of nodes generated and the amount of CPU time consumed. When evaluating the relative performance of algorithms on a single domain, EES frequently comes in second place; however, the difference in performance between it and the best algorithm for a domain is typically very small. In contrast, the other algorithms fail spectacularly in some domains, having performance many times worse than the best algorithm. This makes their aggregate performance poor, as seen in Table 1. While EES is rarely the best algorithm on a single domain, it is the superior algorithm when running on a set of problems with diverse properties or when running on a novel domain.

Discussion

There are two important ideas that lead to the creation of EES. The first is that a lower bound is inappropriate for search guidance except when optimal solutions are absolutely required. This is well recognized for greedy search, where more informed inadmissible heuristics are commonly employed, but has not yet seen wide use in bounded suboptimal search, largely because until now there were no techniques designed to make use of these inadmissible estimates. Some techniques, such as optimistic search, decouple finding solutions and proving bounds, but they were designed to use lower bounds for guidance and bounding, and generally they fail to take advantage of distance to go information. EES does both, and outperforms previous techniques as a direct result.

Conclusions

EES exploits additional information in bounded suboptimal search, resulting in better search orders and shorter solving times across a wide variety of benchmark domains. Unlike previous approaches, explicit estimation search (EES) converts the stated goal of bounded subpotimal search rather directly into an expansion order by taking advantage of inadmissible cost to go and search distance estimators that attempt to be unbiased rather than lower bounds. It is exactly this decoupling of guidance and bounding that allows for the improved performance of EES.

Acknowledgments

We gratefully acknowledge support from NSF (grant IIS-0812141) and the DARPA CSSG program (grant HR0011-09-1-0021).

References

Ghallab, M., and Allard, D. 1983. A_ϵ: An efficient near admissible heuristic search algorithm. In *Proceedings 8th IJCAI*.

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.

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.

Reese, B. 1999. AlphA*: An ϵ-admissible heuristic search algorithm. Unpublished, retrieved from http://home1.stofanet.dk/breese/papers.html.

Samadi, M.; Felner, A.; and Schaeffer, J. 2008. Learning from multiple heuristics. In *Proceedings of AAAI-08*.

Thayer, J. T., and Ruml, W. 2008. Faster than weighted A*: An optimistic approach to bounded suboptimal search. In *Proceedings of the Eighteenth International Conference on Automated Planning and Scheduling (ICAPS-08)*.

Thayer, J., and Ruml, W. 2009. Using distance estimates in heuristic search. In *Proceedings of ICAPS-2009*.

Thayer, J., and Ruml, W. 2010. Finding acceptable solutions faster using inadmissible information. Technical Report 10-01, University of New Hampshire.

Thayer, J. T.; Ruml, W.; and Bitton, E. 2008. Fast and loose in bounded suboptimal heuristic search. In *Proceedings of the First International Symposium on Search Techniques in Artificial Intelligence and Robotics (STAIR-08)*.

Thayer, J.; Ruml, W.; and Kreis, J. 2009. Using distance estimates in heuristic search: A re-evaluation. In *Symposium on Combinatorial Search*.