Heuristic Search Under Quality and Time Bounds

Jordan T. Thayer
Department of Computer Science
University of New Hampshire
Durham, NH 03824 USA
jtd7 at cs.unh.edu

Abstract

Heuristic search is a central component of many important applications in AI including automated planning. While we can find optimal solutions to heuristic search problems, doing so may take hours or days. For practical applications, this is unacceptably slow, and we must rely on algorithms which find solutions of high, but not optimal, quality or ones which bound the time used directly. In my dissertation, I present and analyze algorithms for the following settings: quality bounded heuristic search and time bounded heuristic search. The central theme of my doctoral work will be that taking advantage of additional information can improve the performance of heuristic search algorithms.

1 The Problem

Heuristic search is used to solve a wide variety of problems, such as pathfinding in a grid or temporal planning. When sufficient resources are available, optimal solutions can be found using A* search with an admissible heuristic [Hart *et al.*, 1968]. However if our heuristics contain any error, finding these optimal solutions can require time exponential in the length of the solution [Helmert and Röger, 2008]. Unfortunately our heuristics are frequently inaccurate, making optimal solving prohibitively expensive for many practical applications. Here, a search which finds solutions of reduced quality while consuming fewer resources is preferable.

In these practical settings we must accept suboptimal solutions in order to reduce the cost of finding them. When abandoning solution optimality we still need ways of retaining control over the search. We can choose to bound the quality of the solution returned by an algorithm or we can choose to limit the time used in finding the solution. When bounding solution quality our aim then becomes to solve problems as quickly as possible while guaranteeing that the solutions found have a cost within a bounded factor of optimal. When bounding time, our goal is to produce the best possible solution within the deadline. My dissertation addresses these problems in three parts: first bounded quality, then unknown deadlines, and finally known deadlines.

2 Bounding Solution Quality

First, I consider the setting in which one wants the fastest possible search where the solution is guaranteed to be within a bounded factor of the optimal solution. For a given factor w, we call an algorithm w-admissible if it is guaranteed to return a solution that is no more than w times more expensive than an optimal solution. Weighted A* is an elegant answer to this problem [Pohl, 1970]. In weighted A* the traditional node evaluation function of A*, $f(n) = g(n)+h(n)$, is modified to place additional emphasis on the heuristic evaluation function h, as in $f(n) = g(n) + w \cdot h(n)$.

While previous approaches tend to exploit a single source of information, typically an admissible, or strictly conservative, estimate of the cost remaining, my work focuses on using many heuristics to guide search effectively, including inadmissible estimates [Thayer *et al.*, 2008], estimates of solution length [Thayer and Ruml, 2009], and the average case performance of bounded suboptimal search algorithms [Thayer and Ruml, 2008]. These sources of information have either been totally ignored in the bounded suboptimal setting, or produce fragile behavior.

Inadmissible Heuristics Inadmissible heuristics provide an estimate of the cost to go from a current node to the goal. They cannot provide a tight bound on the quality of solutions. If we separate the roles search guidance and providing lower bounds, and use a separate heuristic for each, we can then use arbitrary inadmissible heuristic for bounded suboptimal search without losing the suboptimality bounds [Thayer and Ruml, 2011b]. Automatically deriving these inadmissible heuristics from the error observed in a base heuristic during the search allows us to tackle novel domains immediately and learn instance specific heuristics.

Distance Estimates In some domains all actions have the same cost, typically these are called unit-cost problems. In many practical domains, actions have varied costs, and so it is useful to separate the idea of pursuing solutions which are cheap and pursuing those which are close. Searches that consider both the estimated cost-to-go as well as the search-distance-to-go have an advantage over searches which do not make use of this additional information. Although there were several proposals in the 70's and 80's [Pearl and Kim, 1982; Pohl, 1973], using distance estimates has fallen out of favor, in part because of the poor performance of previous algorithms. The root of this poor performance has been identified [Thayer *et al.*, 2009] and corrected for in a new algorithm [Thayer and Ruml, 2011a].

Exploit Average Case Performance
Previous approaches to suboptimal heuristic search are very strict in that no node is ever expanded which could not lead to a w-admissible goal. We could instead take an optimistic approach [Thayer and Ruml, 2008] in which we expand nodes more aggressively, without a strict guarantee that they lead to a w-admissible solution. Any fast search method with an open list can be used. Crucially, once we have found a solution using the aggressive search, we then expand additional nodes until we prove our solution is w-admissible.

3 Bounding Solving Time

I will consider the problem of finding solutions under time pressure. Here, the user is constrained by time and not by solution quality, although we'd like to find the best solution possible within the deadline. These problems can be further divided based on whether the deadline is known or not.

Unknown Deadlines Anytime heuristic search algorithms such as anytime heuristic search [Hansen and Zhou, 2007], ARA* [Likhachev *et al.*, 2003], and restarting weighted A* [Richter *et al.*, 2009] address the problem of finding solutions to problems within an unknown deadline. These algorithms use one or more weighted A* searches to find an initial solution and then return a stream of ever improving solutions.

We can use the improved bounded suboptimal algorithms in place of weighted A* in any of these anytime algorithms. If the underlying algorithms are better than weighted A* we would expect the anytime versions of these algorithms to outperform anytime searches relying on previous state of the art techniques. Preliminary results suggest that this is indeed the case [Thayer and Ruml, 2010].

Fixed Deadlines Unlike anytime search algorithms, which are designed to handle unknown time bounds, deadline search algorithms address problems where the time available to find a solution is fixed in advance. Although an anytime algorithm can obviously be used to solve a problem within a fixed deadline, the deadline itself is a useful piece of information that a standard anytime algorithm would ignore. Anytime algorithms find many solutions, only one of which we would ever use. As a result, some of the effort of those algorithms is wasted. By creating deadline aware algorithms we can avoid this wasteful behavior, and we will be able to return higher quality solutions within the same deadline as a result.

4 Conclusions

Although the prevailing sentiment is that the problem of bounded suboptimal search has been solved, leading algorithms do not directly address the issue of solving problems within the bound as quickly as possible; the loosening of the suboptimality bound simply has the pleasant side effect of reducing the time needed to find a solution.

Improvements to bounded suboptimal search extend directly to anytime search, by virtue of the fact that bounded suboptimal searches can be easily converted into anytime search algorithms. Improvements in anytime search will help better address the problem of search under an unspecified deadline. They are not the best approach to the known deadline case. Solving for known deadlines requires search algorithms that react to the approaching deadline. This requires the development of new techniques that measure an algorithms progress and estimated effort to a solution.

5 Acknowledgments

We gratefully acknowledge support from NSF (grant IIS-0812141) and the DARPA CSSG program (grant N10AP20029).

References

[Hansen and Zhou, 2007] Eric A. Hansen and Rong Zhou. Anytime heuristic search. *JAIR*, 28:267–297, 2007.

[Hart *et al.*, 1968] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions of Systems Science and Cybernetics*, SSC-4(2):100–107, July 1968.

[Helmert and Röger, 2008] Malte Helmert and Gabriele Röger. How good is almost perfect? In *Proceedings of AAAI-2008*, 2008.

[Likhachev *et al.*, 2003] Maxim Likhachev, Geoff Gordon, and Sebastian Thrun. ARA*: Anytime A* with provable bounds on suboptimality. In *Proceedings of the Seventeenth Annual Conference on Neural Information Porcessing Systems (NIPS-03)*, 2003.

[Pearl and Kim, 1982] Judea Pearl and Jin H. Kim. Studies in semi-admissible heuristics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-4(4):391–399, July 1982.

[Pohl, 1970] Ira Pohl. Heuristic search viewed as path finding in a graph. *Artificial Intelligence*, 1:193–204, 1970.

[Pohl, 1973] Ira Pohl. The avoidance of (relative) catastrophe, heuristic competence, genuine dynamic weighting and computation issues in heuristic problem solving. In *Proceedings of IJCAI-73*, pages 12–17, 1973.

[Richter *et al.*, 2009] Silvia Richter, Jordan Thayer, and Wheeler Ruml. The joy of forgetting: Faster anytime search via restarting. In *Proceedings of SoCS-09*, 2009.

[Thayer and Ruml, 2008] Jordan T. Thayer and Wheeler Ruml. Faster than weighted A*: An optimistic approach to bounded suboptimal search. In *Proceedings of ICAPS-08*, 2008.

[Thayer and Ruml, 2009] Jordan Thayer and Wheeler Ruml. Using distance estimates in heuristic search. In *Proceedings of ICAPS-2009*, 2009.

[Thayer and Ruml, 2010] Jordan Thayer and Wheeler Ruml. Anytime heuristic search: Frameworks and algorithms. In *Proceedings of SoCS-10*, July 2010.

[Thayer and Ruml, 2011a] Jordan Thayer and Wheeler Ruml. Bounded suboptimal search: A direct approach using inadmissible estimates. In *Proceedings of IJCAI-11*, 2011.

[Thayer and Ruml, 2011b] Jordan Thayer and Wheeler Ruml. Learning inadmissible heuristics during search. In *Proceedings of ICAPS-11*, 2011.

[Thayer *et al.*, 2008] Jordan T. Thayer, Wheeler Ruml, and Ephrat Bitton. Fast and loose in bounded suboptimal heuristic search. In *Proceedings of STAIR-08*, Summer 2008.

[Thayer *et al.*, 2009] Jordan Thayer, Wheeler Ruml, and Jeff Kreis. Using distance estimates in heuristic search: A re-evaluation. In *Proceedings of SoCS-09*, 2009.