Avoiding Re-expansions in Suboptimal Best-First Search

Jingwei Chen,[1] **Nathan R. Sturtevant** [1]
[1]Department of Computing Science, Alberta Machine Intelligence Institute (Amii), University of Alberta, Canada
jingwei5@ualberta.ca, nathanst@ualberta.ca

Abstract

This paper gives a summary of published conditions needed for a priority function to return bounded-optimal solutions when not performing re-expansions of previously expanded states in best-first search.

Introduction

While Weighted A* (Pohl 1970) is a well-studied algorithm for suboptimal search, the question of whether Weighted A* should re-expand states when a shorter path is found has not always been clearly described in the literature. The original paper does not require updating the cost of states already found on open, much less re-expanding them when a shorter path is found. However, later work (Pearl 1984) describes A* as always re-opening states, and then Weighted A* as A* with a different priority function, implying that it would re-open and re-expand states if shorter paths were found. Later work (Likhachev, Gordon, and Thrun 2003; Ebendt and Drechsler 2009) clarifies that Weighted A* does not need to re-open states to find bounded-optimal solutions.

Our recent work (Chen and Sturtevant 2021) characterized the necessary and sufficient conditions for a priority function to find solutions with bounded-optimal cost when states are not re-expanded in best-first search. This research abstract summarizes these results.

Necessary and Sufficient Conditions

Assume that a best-first search is guided by the priority function $f(u) = \Phi(h(u), g(u))$, where $\Phi(x, y)$ is a continuous function $\mathbb{R}^2 \rightarrow \mathbb{R}$. We assume that the goal is to find a solution path with cost $\leq B(C^*)$, where $B : \mathbb{R} \rightarrow \mathbb{R}$ is a given bounding function that satisfies $\forall x \geq 0, B(x) \geq x$.

Φ describes a surface. States on the surface with the same value of Φ have the same priority. We can visualize these states by isolines on the surface of Φ, as in Figure 1.

Then, we assume that Φ has the following properties:

Property 1 *For any given* $\delta > 0$, $\Phi(x + \delta, y) > \Phi(x, y)$, $\Phi(x, y + \delta) > \Phi(x, y)$.

Property 2 *For any given* $\delta > 0$, $\Phi(x, y + \delta) \leq \Phi(x + \delta, y)$

Property 1 requires that if there are two states with same h-cost, the one with lower g-cost will have lower priority. For two states with same g-cost, the one with lower h-cost will have lower priority.

Property 2 requires that if two states have the same $g + h$, the one with lower g-cost will not have lower priority than the one with higher g-cost, although they could have equal priority (as in A*).

Property 3 $\Phi(x, 0) = x$.

This property just sets the scale for Φ. Other scales could be used, but this particular property gives Φ semantic meaning: Φ is a lower bound on the optimal solution cost through a particular state. *We always assume that Properties 1 to 3 hold.* Given these properties, we can then relate the priority function to the bounding function used in two ways.

Property 4 $\Phi(x, 0) \leq \Phi(0, B(x))$*, where* B *is the given bounding function.*

Property 5 $\Phi(x, 0) = \Phi(0, B(x))$*, where* B *is the given bounding function.*

Finally, we define the Φ-inequality as $\Phi(h(n), g(n)) \leq g^*(n) + h(n)$ for some state n.

If the Φ-inequality holds for each state when it is expanded, then a best-first search without state re-expansions will return a solution that is bounded by B. The conditions which are required for the Φ-inequality to hold include:

1. If the heuristic is inconsistent, the Φ-inequality is not guaranteed to hold, and thus re-expansions are necessary.
2. If the heuristic is consistent, Property 6 (below) is both a necessary and sufficient condition for best-first search to find bounded-optimal solutions without re-expansions when Φ meets Property 5.
3. If the heuristic is consistent, and if Φ only meets Property 4, then Property 6 (below) is sufficient but not necessary for best-first search to avoid re-expansions.

A heuristic is consistent on an undirected graph if $\forall m, n, |h(n) - h(m)| \leq d(n, m)$. If this property holds on a directed graph, we say the heuristic is *strongly consistent*. A heuristic is then *weakly consistent* if $\forall m, n, h(n) \leq d(n, m) + h(m)$. Otherwise the heuristic is inconsistent. The following property is analogous to consistency of the heuristic. We refer to this as the Φ function being consistent.

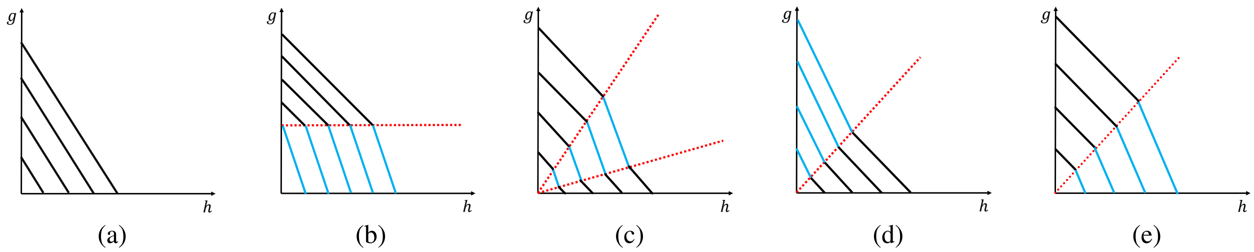

Figure 1: The iso-lines of different Φ functions. (a) Φ_{WA^*} (b) Φ_{AB} (c) Φ_z (from (Chen and Sturtevant 2021)) (d) Φ_{pwXD} (e) Φ_{pwXU}. Red lines indicate free parameters.

Property 6 *For a strongly consistent heuristic: for any given $\delta > 0$, $\Phi(x + \delta, y + \delta) \leq \Phi(x, y) + 2\delta$*

For a weakly consistent heuristic: for any given $\delta > 0$, $\Phi(x + \delta, y) \leq \Phi(x, y) + \delta$

Stated informally, as long as Φ does not change too quickly, a best-first search will find bounded-optimal solutions without re-expansions.

Priority Functions

Some priority functions have been previously described (Chen and Sturtevant 2019); this work allows a broader range of priority functions, including piecewise functions.

First, note that with these conditions, Weighted A*, when written as $f = g + wh$ (or $\Phi(x, y) = y + wx$) does not meet the conditions required to avoid re-expansions, but this is only because of Property 3. Weighted A* is estimating the cost of the solution that will be found, not the optimal solution. When re-written as $f = g/w + h$ it performs identically and meets all conditions necessary to avoid re-expansions. We now describe several other functions:

$$\Phi_{AB}(x, y) = \begin{cases} x + \frac{K-\gamma}{K} y & y < K \\ x + y - \gamma & y \geq K \end{cases} \tag{1}$$

This priority function is defined for $B_\gamma(x) = x + \gamma$, where γ is a given constant (Valenzano et al. 2013). We recommend setting $K = max\{h(start), \gamma + 1\}$.

For linear bounds, possible functions include

$$\Phi_{pwXD}(x, y) = \begin{cases} y + x & y < \frac{K-w}{w-1} x \\ \frac{1}{w}(y + Kx) & \frac{K-w}{w-1} x \leq y \end{cases} \tag{2}$$

$$\Phi_{pwXU}(x, y) = \begin{cases} \frac{1}{K} y + x & y < \frac{K(w-1)}{K-w} x \\ \frac{1}{w}(y + x) & \frac{K(w-1)}{K-w} x \leq y \end{cases} \tag{3}$$

K is a free parameter, but we recommend setting $K = 2w - 1$. For pwXD (Equation 4), $K > 2w - 1$ violates Property 6. Re-writing Φ_{pwXD} as a functions of f gives:

$$f_{pwXD}(n) = \begin{cases} g(n) + h(n) & g(n) < h(n) \\ \frac{1}{w}(g(n) + (2w - 1)h(n)) & h(n) \leq g(n) \end{cases} \tag{4}$$

A selection of functions are illustrated in Figure 1. Of particular note is f_{pwXD}, which begins by searching optimally (the isolines have slope of -1), and then switches to searching with a weight of $(2w - 1)$. When the number of states with $g(n) < h(n)$ is small, f_{pwXD} is able to search with a larger weight through much of the search, and thus can have very good performance. These, and other functions, are available in the literature (Chen and Sturtevant 2019, 2021).

Acknowledgments

This work was funded by the Canada CIFAR AI Chairs Program. We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

Chen, J.; and Sturtevant, N. R. 2019. Conditions for Avoiding Node Re-expansions in Bounded Suboptimal Search. *International Joint Conference on Artificial Intelligence (IJCAI)* .

Chen, J.; and Sturtevant, N. R. 2021. Necessary and Sufficient Conditions for Avoiding Reopenings in Best First Suboptimal Search with General Bounding Functions. *AAAI Conference on Artificial Intelligence* .

Ebendt, R.; and Drechsler, R. 2009. Weighted A? search–unifying view and application. *Artificial Intelligence* 173(14): 1310–1342.

Likhachev, M.; Gordon, G. J.; and Thrun, S. 2003. ARA*: Anytime A* with provable bounds on sub-optimality. *Advances in neural information processing systems* 16: 767–774.

Pearl, J. 1984. *Heuristics: intelligent search strategies for computer problem solving*. Addison-Wesley Pub. Co., Inc., Reading, MA.

Pohl, I. 1970. Heuristic search viewed as path finding in a graph. *Artificial intelligence* 1(3-4): 193–204.

Valenzano, R.; Arfaee, S. J.; Stern, R.; Thayer, J.; and Sturtevant, N. 2013. Using Alternative Suboptimality Bounds in Heuristic Search. In *International Conference on Automated Planning and Scheduling (ICAPS)*, 233–241.