Alternative Forms of Bounded Suboptimal Search

Richard Valenzano, Shahab Jabbari Arfaee
Department of Computing Science
University of Alberta
{valenzan,jabbaria}@cs.ualberta.ca

Jordan Thayer
Department of Computer Science
University of New Hampshire
jtd7@cs.unh.edu

Roni Stern
ISE Department
Ben Gurion University
roni.stern@gmail.com

Finding optimal solutions to search problems is often not practical due to run-time constraints. This has led to the development of algorithms that sacrifice solution quality in favour of a faster search. The solution quality of a *suboptimal* solution is often measured by the ratio $\frac{C}{C^*}$, where C is the cost of the solution found and C^* is the cost of the optimal solution. Some search algorithms allow a user to define an acceptable level of solution suboptimality that any solution returned must adhere to. Such algorithms are referred to as *bounded suboptimal* algorithms. For example, Weighted A* (WA*) (Pohl 1970) guarantees that the ratio $\frac{C}{C^*}$ of any solution found is at most $(1+\epsilon)$, where ϵ is the acceptable level of suboptimality defined by the user.

General Bounding

While most current bounded suboptimal algorithms measure suboptimality as the ratio $\frac{C}{C^*}$, this is not the only possible way to measure it. For example, a natural alternative is to require that any solution found is no larger than $C^* + \gamma$ for some constant $\gamma \geq 0$. This paradigm, which we will refer to as *additive bounding*, corresponds to the use of $C - C^*$ as a suboptimality measure. Other possible bounding paradigms include requiring that $C \leq (C^*)^p$ for some $p \geq 1$, or requiring that $C \leq a^{C^*}$ for some $a > 1$. These correspond to the suboptimality measures of $\log_{C^*} C$ and C^{1/C^*}, respectively.

The notion of a bounding paradigm can be generalized through the use of a *bounding function*, defined as follows:

Definition 1 *An algorithm A is said to satisfy a given function $B : \mathbb{R} \to \mathbb{R}$ where $B(x) \geq x$ for all x, if the cost of any solution found by A is bounded by $B(C^*)$. The function B is referred to as the bounding function of A.*

For example, the bounding function satisfied by Weighted A* is $B_\epsilon(x) = (1+\epsilon) \cdot x$ and the bounding function for additive bounded algorithms is $B_\gamma(x) = x + \gamma$.

In this paper, we present several methods for constructing bounded suboptimal search algorithms for a given bounding function. We will first consider doing so in the best-first search and iterative deepening search frameworks. These approaches use an evaluation function to guide their search, and the following theorem identifies how to select one so as to satisfy a given bounding function:

Theorem 1 *For a given bounding function B, if there exists an evaluation function F such that $F(n) \leq B(g(n) + h^*(n))$ for any node n, then a best-first or iterative deepening search that is guided by F will satisfy B.*

Note, all proofs are omitted due to space constraints.

This theorem offers a sufficient condition on the evaluation function used in a best-first or iterative deepening search so as to satisfy a given bounding function. Of particular note, if a bounding function B is such that $B(x+y) \geq x + B(y)$ for all x and y, then the theorem identifies a way in which we can immediately construct algorithms that satisfy B. To do so, simply use the evaluation function $F_B(n) = g(n) + B(h(n))$ to guide a best-first or iterative deepening search. This is the approach taken by WA*, for example.

A similar theorem can be proven for *focal list-based search* algorithms (denoted as focal algorithms) like A^*_ϵ (Pearl and Kim 1982) or EES (Thayer and Ruml 2011). In every iteration of a focal algorithm, a subset of nodes from the open list (OPEN), called the *focal list* (FOCAL), are selected as candidates for expansion. A given policy is then used to select a node from the focal list to be expanded next. In ϵ-admissible focal algorithms, FOCAL contains any node n for which $f(n) \leq (1+\epsilon) \cdot f(n_{best})$, where $f = g + h$, h is admissible, and n_{best} is the node on OPEN with the lowest value of f. By changing the way that FOCAL is defined, these algorithms can be set to satisfy a given strictly increasing bounding function as follows:

Theorem 2 *For a given strictly increasing bounding function B, if a focal algorithm defines its focal list to include any node n such that $f(n) \leq B(f(n_{best}))$, then this focal algorithm will satisfy B, regardless of the policy used to select nodes from the focal list.*

Alternative evaluation functions and FOCAL definitions have previously been considered by Dechter and Pearl (1985), as well as by Farreny (Farreny 1999). These works focus on determining suboptimality bounds for a given evaluation function or FOCAL definition. While those works can be used to determine the suboptimality level achievable with a given evaluation function or FOCAL definition, they did not aim to develop algorithms for a given bounding paradigm, which is the goal of this research.

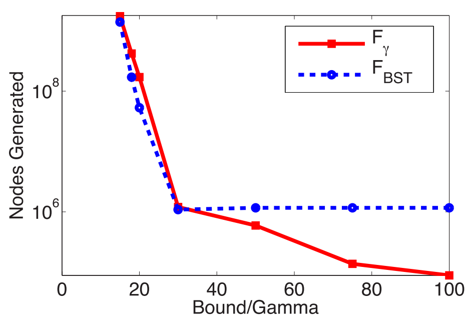

Figure 1: Iterative Deepening on the 24-puzzle.

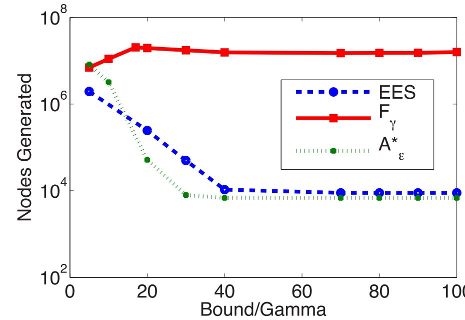

Figure 2: Focal Algorithms on the Inverse 15-puzzle.

Additive Bounding

Next, we use the theorems above to construct search algorithms with an additive bound. Following Theorem 1, one might propose the use of the evaluation function $F(n) = g(n) + h^*(n) + \gamma$. However, it is easy to see that a best-first search guided by this function will behave exactly like A*. This means that there will be no speedup over A* when using this evaluation function, thus defying the reason for even allowing for suboptimal solutions in the first place.

Instead, we propose the following function which can be used to speed up search and still satisfy an additive bound:

$$F_\gamma(n) = g(n) + h(n) + \gamma \cdot \min(h(n)/h(n_i), 1)$$

where n_i is the initial node and h is an admissible heuristic function. As $F_\gamma(n) \leq g(n) + h^*(n) + \gamma$ clearly holds, searching with this function will satisfy the additive bounding function $B_\gamma(x) = x + \gamma$ by Theorem 1.

If we also have an inadmissible heuristic h_i in addition to the admissible h, then the following evaluation function will also satisfy the additive bounding function by Theorem 1:

$$F'_\gamma(n) = g(n) + \min(h_i(n), h(n) + \gamma)$$

This evaluation function is notable as it allows us to search with guidance from an inadmissible heuristic while still satisfying an additive bound, which would not be possible if the inadmissible heuristic were used alone.

Experimental Evaluation

To demonstrate the effectiveness of these evaluation functions, we ran experiments with an iterative deepening search on 50 24-puzzle problems. Results are shown in Figure 1. The x-axis shows the additive bound used (γ) and the y-axis shows the average number of nodes generated to solve the instances (in log scale). The admissible heuristic used was the state-of-the-art 6-6-6-6 PDB (Korf and Felner 2002), while the inadmissible heuristic used was the bootstrapping heuristic developed by Jabbari Arfaee et. al. (2011).

The figure shows that using F_γ can effectively sacrifice guaranteed solution quality for improved search speed in a similar manner as does WA*. Additionally, F_{BST} (which corresponds to F'_γ when using the bootstrapping heuristic) is also shown to outperform F_γ until $\gamma \geq 50$. As $h_i(n) \leq h(n) + 50$ for all n, any search guided by F_{BST} with $\gamma \geq 50$ will be equivalent to a search guided by $F = g + h_i$. This is unlike a search guided by F_γ which will become increasingly greedy even as γ increases beyond 50.

When all actions are not of equal cost, BFS and iterative techniques are generally less effective than the focal-based approaches, which can more easily employ distance-to-go heuristics (Thayer and Ruml 2011). Fortunately, Theorem 2 describes a simple way to adapt these ϵ-admissible focal algorithms to satisfy an additive bound.

These new additive algorithms, along with a best-first search guided by F_γ, were tested on 100 instances of the inverse 15-puzzle problems (Thayer and Ruml 2011). The results of this experiment are shown in Figure 2. The admissible heuristic used is the cost-based Manhattan distance, while the Manhattan distance ignoring action costs is used for the distance-to-go heuristic.

As seen in the figure, the focal algorithms both effectively sacrifice guaranteed solution quality for improved search time. They are also significantly outperforming the best-first search guided by F_γ, which is much less effective in this domain than in unit-cost domains. This is consistent with the behaviour seen when comparing WA* to the standard ϵ-admissible versions of A^*_ϵ and EES.

Acknowledgments

This research was supported by NSERC and iCore.

References

Dechter, R., and Pearl, J. 1985. Generalized Best-First Search Strategies and the Optimality of A*. *J. ACM* 32(3):505–536.

Farreny, H. 1999. Completeness and Admissibility for General Heuristic Search Algorithms-A Theoretical Study: Basic Concepts and Proofs. *J. Heuristics* 5(3):353–376.

Jabbari Arfaee, S.; Zilles, S.; and Holte, R. C. 2011. Learning heuristic functions for large state spaces. *Artificial Intelligence* 175(16-17):2075–2098.

Korf, R. E., and Felner, A. 2002. Disjoint pattern database heuristics. *Artificial Intelligence* 134:9–22.

Pearl, J., and Kim, J. H. 1982. Studies in Semi-Admissible Heuristics. *IEEE Trans. on Pattern Recognition and Machine Intelligence* 4(4):392–399.

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

Thayer, J. T., and Ruml, W. 2011. Bounded Suboptimal Search: A Direct Approach Using Inadmissible Estimates. In *IJCAI*, 674–679.