Bidirectional Bounded-Suboptimal Heuristic Search with Consistent Heuristics (Extended Abstract)

Shahaf S. Shperberg[1], **Natalie Morad**[1], **Lior Siag**[1], **Ariel Felner**[1], **Dor Atzmon**[2],
[1] Department of Software and Information Systems Engineering, Ben-Gurion University, Israel
[2] Department of Computer Science, Bar-Ilan University, Israel
{shperbsh,felner}@bgu.ac.il, {moradna,siagl}@post.bgu.ac.il, dor.atzmon@biu.ac.il

Introduction

Recent advancements in bidirectional heuristic (BiHS) search have yielded significant theoretical insights and novel algorithms. While most previous work has concentrated on optimal search methods, this work focuses on *Bounded-suboptimal search* (BSS), where a bound on the suboptimality of the solution cost is specified. We build upon the state-of-the-art optimal bidirectional search algorithm, BAE* (Sadhukhan 2013), designed for consistent heuristics, and introduce a family of algorithms, called WBAE*, that adapt BAE* into the BSS.

Background

In BiHS, the aim is to find a least-cost path, of cost C*, between $start$ and $goal$ in a given graph G. $c(x, y)$ denotes the cost of the cheapest path between x and y, so $c(start, goal) = C^*$. BiHS executes a forward search (F) from $start$ and a backward search (B) from $goal$ until the two searches meet. BiHS algorithms typically maintain two open lists OPEN_F and OPEN_B for the forward and backward searches, respectively. Each node has a g-value, an h-value, and an f-value (g_F, h_F, f_F and g_B, h_B, f_B for the forward and backward searches, correspondingly). For a direction D (F or B), f_D, g_D, and h_D represent the f-, g-, and h-values in that direction. We use $xMin_D$ to denote the minimal x value in OPEN_D, e.g., $gMin_F$ represents the minimal g-value in OPEN_F. Most BiHS algorithms consider the two *front-to-end* heuristic functions $h_F(s)$ and $h_B(s)$ which respectively estimate $c(s, goal)$ and $c(start, s)$ for all $s \in G$. h_F is *forward admissible* iff $h_F(s) \leq c(s, goal)$ for all s in G and is *forward consistent* iff $h_F(s) \leq c(s, s') + h_F(s')$ for all s and s' in G. Backward *admissibility* and *consistency* are defined analogously.

BAE* (Sadhukhan 2013) is a BiHS algorithm that assume heuristic consistency. Let $d_F(n) = g_F(n) - h_B(n)$, the *difference* between the actual forward cost of n (from $start$) and its heuristic estimation to $start$. This indicates the *heuristic error* for node n (as $h_B(n)$ is a possibly inaccurate estimation of $g_F(n)$). Likewise, $d_B(m) = g_B(m) - h_F(m)$. BAE* orders nodes in OPEN_F according to:

$$b_D(n) = g_D(n) + h_D(n) + (g_D(n) - h_{\overline{D}}(n)) = f_D(n) + d_D(n)$$

where $h_{\overline{D}}$ is the heuristic in the direction opposite of D. $b_D(n)$ adds the heuristic error $d_D(n)$ to $f_D(n)$ to account for the underestimation by $h_{\overline{D}}(n)$. During each expansion cycle, BAE* chooses a search direction D and expands a node with minimal b_D-value. Additionally, BAE* terminates once the same state n is found on both open lists and the cost of the path from $start$ to $goal$ through n is $\leq LB_B$ where LB_B, is a the following lower bound on C^*:

$$LB_B = (bMin_F + bMin_B)/2 \tag{1}$$

where $bMin_D$ is the minimal b-value in OPEN_D.

Given a consistent heuristic, BAE* was proven to return an optimal solution. $b(n)$ is more informed than other priority functions (as it also considers $d(n)$). BAE* was shown to outperform common unidirectional and bidirectional algorithms on different domains, with improvements reported of up to an order of magnitude. Therefore, BAE* is considered a state-of-the-art BiHS algorithm for consistent heuristics.

Weighted BAE*

We present a BSS version of BAE* that uses the idea of weighted A* (WA*) (Pohl 1970) of inflating h by a user-provided parameter $W \geq 1$ and ensures that the cost of the returned solution is bounded by $W \cdot C^*$. Consider again the BAE* formula for prioritizing nodes:

$$b_D(n) = g_D(n) + h_D(n) + d_D(n)$$

In WA*, the h-value is multiplied by a factor of W. The core strategy of WA* is to prioritize nodes with lower h-values, effectively guiding the search towards paths with seemingly lower estimated costs, while preserving the suboptimality bounds. This mechanism is a crucial feature that we preserve in our approach (and was similarly preserved by WMM). The main challenge is to determine how to address the heuristic error d_D, an aspect of BAE* that was never before considered in a BSS setting.

We propose the following formula which provides a range of possibilities of how to weigh the heuristic error $d_D(n)$ in the weighted version of BAE*, which we denote as WBAE*:

$$b_{W_D}(n) = g_D(n) + W \cdot h_D(n) + \lambda \cdot d_D(n) \tag{2}$$

where $\lambda \leq W$ is a real number. There are several special cases to consider. If $\lambda = 0$, then $b_{W_F}(n)$ is similar to the

Algorithm	TOH-12 (10+2) 1	1.1	1.2	1.5	1.7	2	3	5	10
WA*	279K	137K	54K	4K	2K	1K	417	433	441
WBiA*	236K	106K	40K	**3K**	**1K**	**627**	**374**	**343**	347
WMM	337K	164K	60K	3K	2K	2K	5K	4K	556
WBS*	179K	82K	33K	**3K**	**1K**	649	377	349	353
WBAE* $\frac{1}{W^2}$	**48K**	30K	**17K**	5K	3K	1K	394	353	**343**
WBAE* $\frac{1}{W}$	**48K**	**28K**	18K	7K	4K	1K	405	364	**343**
WBAE* 1	**48K**	**28K**	20K	9K	6K	3K	536	380	357
WBAE* W	**48K**	29K	23K	13K	11K	7K	3K	2K	1K

Algorithm	TOH-12 (8+4) 1	1.1	1.2	1.5	1.7	2	3	5	10
WA*	2M	2M	1M	486K	318K	278K	26K	13K	20K
WBiA*	1M	806K	558K	186K	109K	39K	**11K**	**5K**	**4K**
WMM	1M	863K	707K	269K	157K	56K	15K	10K	8K
WBS*	924K	656K	466K	167K	94K	35K	**11K**	6K	**4K**
WBAE* $\frac{1}{W^2}$	**189K**	182K	162K	105K	**74K**	**34K**	12K	**5K**	**4K**
WBAE* $\frac{1}{W}$	**189K**	162K	137K	**97K**	76K	42K	14K	6K	**4K**
WBAE* 1	**189K**	148K	**130K**	105K	92K	69K	22K	9K	5K
WBAE* W	**189K**	**143K**	131K	109K	102K	88K	71K	56K	41K

Algorithm	TOH-12 (6+6) 1	1.1	1.2	1.5	1.7	2	3	5	10
WA*	3M	3M	3M	2M	2M	1M	1M	863K	331K
WBiA*	2M	2M	1M	682K	439K	266K	117K	72K	**43K**
WMM	964K	1M	1M	842K	600K	395K	251K	121K	64K
WBS*	2M	1M	1M	584K	390K	248K	**115K**	76K	49K
WBAE* $\frac{1}{W^2}$	**382K**	403K	402K	358K	320K	**221K**	119K	**71K**	45K
WBAE* $\frac{1}{W}$	**382K**	362K	342K	310K	284K	227K	130K	77K	47K
WBAE* 1	**382K**	333K	317K	301K	292K	263K	176K	111K	68K
WBAE* W	**382K**	**318K**	**306K**	**282K**	**273K**	252K	227K	202K	186K

Table 1: Average number of node expansions on the 12-disks ToH domain, with $(10+2)$, $(8+4)$, and $(6+6)$ PDBs

priority function used in WBiA*. When $\lambda = W$, $d_F(n)$ is treated as an integral part of the heuristic. For intermediate values where $0 < \lambda < W$, $b_{W_F}(n)$ provides an approach that balances between these extremes.

WBAE* terminates once a state n is found on both open lists and the cost of the path from $start$ to $goal$ through n is $\leq LB_{\texttt{WB}}$, where:

$$LB_{\texttt{WB}} = (b_W Min_F + b_W Min_B)/2 \quad (3)$$

and $b_W Min_D$ is the minimal b_{W_D}-value in $\textsc{Open}_\textsc{D}$.

Given consistent heuristics, WBAE* finds a bounded-suboptimal solution even if never re-expanding nodes.

The Role of λ

In heuristic search, algorithms must balance two tasks: (1) finding and refining solutions (lowering the upper bound U), and (2) proving (sub)optimality (raising the lower bound LB). Termination requires $U \leq LB$, making both tasks essential—yet often in tension.

Using the heuristic error d helps raise LB but can hinder solution finding. Thus, different strategies are needed depending on the challenge. If a strong initial heuristic yields a high LB, the focus should be on quickly finding a solution—favoring small or zero λ, especially when valid solutions are rare. Conversely, when many (sub)optimal solutions exist but are hard to verify, the focus shifts to increasing LB, favoring large λ.

We expect increasing LB to dominate when (i) the heuristic is weak and (ii) the suboptimality bound W is small.

Hypothesis on the role of λ: larger values of λ are more effective when the heuristic is weak and the suboptimality bound W is small. Smaller values of λ are preferable when the heuristic is more accurate or when W is large.

Empirical Evaluation

We compare all existing BiHS BSS algorithms: WBiA* Köll and Kaindl (1993), WBS* Köll and Kaindl (1993), WMM (Atzmon et al. 2023), and WBAE*, along with WA* as a representative UniHS BSS algorithm. For WBAE* we explore various values ($\lambda \in \{0, \frac{1}{W^2}, \frac{1}{W}, 1, W\}$) to investigate the influence of λ on search performance. All algorithms break ties in favor of higher g-values.

Table 1 shows the results on 100 problem instances of the 12-disk *4-peg Towers of Hanoi* (TOH) problem (10+2), (8+4), and (6+6) additive Pattern Databases (PDBs).

For optimal search ($W = 1$), WBAE* (effectively BAE*) significantly outperforms the other algorithms, and in particular, it outperforms WA* (effectively A*) by a factor of 6 to 10, depending on the heuristic. This demonstrates the importance of using the heuristic error d, when available, for proving the optimality of solutions. However, in accordance with our hypothesis, as W increases, the focus of the search shifts from proving optimality to finding a solution. Thus, the heuristic error d becomes less important, and can even hinder the search. For example, with the (10+2) PDB, $\lambda = W$ is never the best policy for $W > 1$ among all the examined values. For $W = 1.1$, we observe that $\lambda = 1$ and $\lambda = \frac{1}{W}$ have roughly the same performance (28K expansions), but for $W = 1.2$, a lower value of $\lambda = \frac{1}{W^2}$ yields the best results. For $W \geq 1.5$, it is better to ignore d altogether and use $\lambda = 0$ (= WBiA*).

For the weaker heuristics, (8+4) and (6+6), the trends are similar, although the transition point between the best values of λ is different. For example, with the (8+4) heuristic, $\lambda = W$ is best performing up to $W = 1.1$, while with the (6+6) heuristic, $\lambda = W$ is better up to $W = 2$. This observation is also in agreement with our hypothesis, as we see that when the heuristic gets weaker, higher values of λ tend to work better. Notably, in this domain, WA* was never the best choice, although the advantage of the BiHS algorithm diminishes as W increases. Moreover, neither WMM nor WBS* emerged as the best algorithms. WBiA* (similar to WBAE* with $\lambda = 0$) outperformed all other variants in some configurations with larger values of W, particularly when using the stronger heuristic (10+2). However, it struggled with smaller values of W, especially when combined with weaker heuristics (8+4 and 6+6). This pattern aligns with our hypothesis: when heuristics are strong and W is large, less effort is needed to validate (sub)optimality, diminishing the impact of λ.

References

Atzmon, D.; Shperberg, S. S.; Sabah, N.; Felner, A.; and Sturtevant, N. R. 2023. W-restrained Bidirectional Bounded-Suboptimal Heuristic Search. In *ICAPS*, 26–30.

Köll, A. L.; and Kaindl, H. 1993. Bidirectional Best-First Search with Bounded Error: Summary of Results. In *IJCAI*, 217–223.

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

Sadhukhan, S. K. 2013. Bidirectional heuristic search based on error estimate. *CSI Journal of Computing*, 2(1-2): S1:57–S1:64.