W-restrained Bidirectional Bounded-Suboptimal Heuristic Search

Dor Atzmon[1], Shahaf S. Shperberg[2], Netanel Sabah[2], Ariel Felner[2], Nathan R. Sturtevant[3]

[1] Department of Computer Science, Royal Holloway, University of London, United Kingdom
[2] Department of Software and Information Systems Engineering, Ben-Gurion University, Israel
[3] Department of Computing Science, Alberta Machine Intelligence Institute (Amii), University of Alberta, Canada
dor.atzmon@rhul.ac.uk, shperbsh@bgu.ac.il, sabahn@post.bgu.ac.il, felner@bgu.ac.il, nathanst@ualberta.ca

Abstract

In this paper, we develop theoretical foundations for bidirectional bounded-suboptimal search (BiBSS) based on recent advancements in optimal bidirectional search. In addition, we introduce a BiBSS variant of the prominent meet-in-the-middle (MM) algorithm, called Weighted MM (WMM). We show that WMM has an interesting property of being W-restrained, and study it empirically.

1 Introduction

In optimal search, the task is to find the least-cost path between two vertices, $start$ and $goal$, in a given graph. *Unidirectional heuristic search* (UHS) admissible algorithms, such as A* (Hart, Nilsson, and Raphael 1968), that order nodes n according to the $f(n) = g(n) + h(n)$ formalization, return an optimal solution (Dechter and Pearl 1985) (of cost C^*), if h is *admissible* (never overestimating).

In many cases, finding optimal solutions is infeasible due to the immense computation required. *Bounded-suboptimal search* (BSS) is a paradigm which trades the quality of the solution for faster running time. In BSS, we are given a bound $W \geq 1$, and are required to find a solution with cost $\leq W \cdot C^*$. A classical BSS algorithm is *Weighted A** (WA*) (Pohl 1970), which expands nodes n according to$f(n) = g(n) + W \cdot h(n)$. Given an admissible heuristic, WA* is guaranteed to return a bounded-suboptimal solution. Other BSS algorithms, include *Dynamic Potential Search* (DPS) (Gilon, Felner, and Stern 2016), Explicit Estimation Search (EES) (Thayer and Ruml 2011), XDP and XUP (Chen and Sturtevant 2019, 2021), and improved versions of these algorithms (Fickert, Gu, and Ruml 2022).

Bidirectional heuristic search (BiHS) is an alternative to UHS that progresses simultaneously from $start$ (forward) and $goal$ (backward) until the two frontiers meet. Recent work presented theoretical study and practical ways for using BiHS with tremendous achievements (Barker and Korf 2015; Holte et al. 2017; Eckerle et al. 2017; Shaham et al. 2017; Chen et al. 2017; Shperberg et al. 2019a,b, 2021; Alcázar, Riddle, and Barley 2020; Alcázar 2021).

Yet, while unidirectional BSS (UBSS) was broadly studied, bidirectional BSS (BiBSS) has barely been explored. This paper aims to build a new foundation for BiBSS and takes a first step towards adapting the recent advancements in BiHS to the BSS setting. First, we extend the theory of *must-expand pairs* (Eckerle et al. 2017) and of Focal Search to BSS. Then, we develop WMM — a W*-restrained* BiBSS algorithm based on the *meet-in-the-middle* (MM) algorithm (Holte et al. 2017). WMM never expands nodes with cost $> W \cdot C^*/2$. WMM is empirically evaluated and cases where it is beneficial are presented and explained.

2 Background on BiHS

In BiHS, two open lists OPEN_F and OPEN_B are maintained for the forward and backward searches, respectively. Each node is associated with 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). MM (Holte et al. 2017) is a BiHS algorithm that guarantees that the search frontiers *meet in the middle*. In MM, nodes n in OPEN_F (respectively OPEN_B) are prioritized by $pr_F(n) = \max(f_F(n), 2g_F(n))$. Let $prmin_F$ and $prmin_B$ be the minimum priority on OPEN_F and OPEN_B. MM expands the node with priority $C = \min(prmin_F, prmin_B)$. Let U (initially set to ∞) denote the incumbent solution cost. When OPEN_F and OPEN_B contain the same node n, a solution is found and we set $U \leftarrow \min(U, g_F(n) + g_B(n))$. Let $LB = \max(C, fmin_F, fmin_B, gmin_F + gmin_B)$, where $gmin_F$, $gmin_B$, $fmin_F$ and $fmin_B$ are the lowest g- and lowest f-values in OPEN_F and OPEN_B, respectively. LB is a lower bound on C^*.[1] Thus, MM halts when $U \leq LB$. Given a fraction $0 \leq p \leq 1$, *Fractional MM* (fMM(p)) (Shaham et al. 2017) extends MM and prioritizes nodes by $pr_F(n) = \max(f_F(n), g_F(n)/p)$ and $pr_B(n) = \max(f_B(n), g_B(n)/(1-p))$. A BiHS algorithm is *restrained* (Shaham et al. 2017) with respect to a fraction $0 \leq p \leq 1$ if the forward- and the backward search never expand a node n with $f_F(n) > p \cdot C^*$ and $f_B(n) > (1-p) \cdot C^*$, respectively. MM is restrained with respect to $p = 1/2$, while fMM(p) is restrained with respect to any fraction p.

Existing BiBSS algorithms. As opposed to the rich literature on UBSS, the work on BiBSS is very limited. Köll and Kaindl (1993) presented BSS variants of two early BiHS

[1] If known, the cost of the least-cost edge can be used throughout the paper in our lower-bound definitions (Shaham et al. 2018).

algorithms, BHPA (Pohl 1971) and BS* (Kwa 1989). A*-connect (Islam, Narayanan, and Likhachev 2016) is another BiBSS algorithm designed for motion planning. A*-connect uses BHPA with an additional inadmissible heuristic that aims to connect the two frontiers quickly.

2.1 Must-Expand and Never-Expand Nodes

Dechter and Pearl (1985) classified nodes in UHS into the following three categories. **(1)** Nodes with $f(n) < C^*$ are called *must expand nodes* (MEN) as any optimal algorithm must expand these nodes to verify the optimality of C^*. **(2)** An algorithm (e.g., A*) is *optimally efficient* if it never expands nodes with $f(n) > C^*$. We denote such nodes as *never expand nodes* (NENs). **(3)** Some nodes with $f(n) = C^*$ may be expanded; this depends on the tie-breaking rule defined by the algorithm. Figure 1(a) shows the three categories.[2] The blue circle contains the MENs and the outer area contains the NENs. We denote the yellow area as the *leeway zone*. It contains nodes with $f(n) = C^*$; some of which are expanded to find a solution. This analysis relies on the following three assumptions. **(1)** Algorithms are deterministic, have no domain knowledge other than the given heuristics, and generate nodes only via expansions (known as the"DXBB" assumption (Eckerle et al. 2017). **(2)** Algorithms are *admissible*, i.e., they must return optimal solutions (if such exist) when given an admissible heuristic. **(3)** The algorithms are running on instances with a *consistent heuristic* (i.e., $h(n) \leq c(p,n) + h(p)$ for all nodes n, p). We make the same three assumptions throughout this paper.

Eckerle et al. (2017) extended the above analysis to BiHS and defined a *Must-Expand Pair* (MEP). Two nodes u and v (from the forward and backward sides, resp.) are a MEP if all of the following three conditions hold: (1) $f_F(u) < C^*$, (2) $f_B(v) < C^*$, and (3) $g_F(u) + g_B(v) < C^*$; this means that either u or v must be expanded by any optimal BiHS algorithm. Also, similarly to UHS, BiHS algorithms should never expand neither forward nodes u with $f_F(u) > C^*$, nor backward nodes v with $f_B(v) > C^*$. The blue circles in Fig. 1(c) show the nodes that are part of MEPs that are expanded by MM, the yellow area is the *leeway zone* (nodes n in direction D with $g_D(n) = C^*/2$ and $f_D(n) = C^*$), and the outer area contains the never-expand nodes.

3 MEPs and NENs in BSS

We now adapt the theory of MEPs and NENs to BSS. For BSS, algorithms do not need to be admissible (assumption 2), but rather be *bounded* (Chen and Sturtevant 2021), i.e., guaranteed to find a solution of cost $\leq W \cdot C^*$ when given a cost-bound W and an admissible heuristic. With this revised assumption 2, we begin by formulating the UBSS case. In UBSS, a node n with $f(n) < C^*/W$ *must be expanded* to verify that the cost of the found solution is below the bound W, when given a consistent heuristic. Likewise, a node n with $f(n) > W \cdot C^*$ should *never be expanded* as no solution with cost $\leq W \cdot C^*$ passes through n. An illustration of the zones is shown in Fig. 1(b). As W increases, the MEN zone is smaller and BSS algorithms spend less time to prove the bound W. Likewise, the *leeway zone* is larger in BSS as it contains all nodes n with $C^*/W \leq f(n) \leq W \cdot C^*$ and such algorithms have more leeway to find a solution. Therefore, increasing W usually decreases the running time.

[2] For clarity, the figures depict brute-force search, in which for every node n, $h(n) = 0$ and thus $g(n) = f(n)$.

3.1 MEPs and NENs in BiBSS

Theorem 1. *A pair of nodes (u, v) is a MEP in BiBSS if all of the following conditions hold: (1) $f_F(u) < C^*/W$, (2) $f_B(v) < C^*/W$, and (3) $g_F(u) + g_B(v) < C^*/W$.*

Proof. By contradiction. Assume that there exists a bounded algorithm A, a problem instance I_1 with optimal solution C_1^*, consistent heuristics h_F and h_B, and a pair of nodes u, v that satisfy the above conditions, but that A does not expand either u or v. We denote the cost of the solution found by A when running on problem I_1 as $A(I_1)$. Since A is bounded, $C_1^* \leq A(I_1) \leq W \cdot C_1^*$. To show the contradiction, another problem instance I_2 can be constructed, which is identical to I_1 with the exception of an edge between u and v with cost $c(u,v) = \max\{h_F(u) - g_B(v), h_B(v) - g_F(u), \frac{C_1^*/W - g_F(v) - g_B(u)}{2}\}$. By construction, the shortest path from $start$ to $goal$ that passes through u and v in I_2 has a cost of $g_F(u) + g_B(v) + c(u,v) = \max\{f_F(u), f_B(v), \frac{C_1^*/W + g_F(v) + g_B(u)}{2}\}$. The first two elements in this max term are strictly smaller than C_1^*/W, due to conditions 1-2, respectively. In addition, since $g_F(u) + g_B(v) < C_1^*/W$ (condition 3), the last element in the max term is also strictly smaller than C_1^*/W, as $\frac{C_1^*/W + g_F(v) + g_B(u)}{2} < \frac{C_1^*/W + C_1^*/W}{2} = C_1^*/W$. Thus, the optimal solution cost of I_2, C_2^*, holds that $C_2^* < C_1^*/W$. In addition, h_F and h_B are admissible on I_2, due to the admissibility and consistency on I_1. Thus, any BSS algorithm must return a solution of cost $\leq W \cdot C_2^* < W \cdot C_1^*/W = C_1^*$. Since A cannot differentiate between I_1 and I_2 without expanding either u or v, A would expand the exact set of nodes when running on I_1 and on I_2, thus $A(I_1) = A(I_2)$. Therefore, we get that $A(I_2) = A(I_1) \geq C_1^* > W \cdot C_2^*$. As a result, A fails to find a bounded-suboptimal solution for I_2, in contradiction to the assumption that A is bounded. □

NENs in BiBSS are defined similarly to UBSS: a node n should never be expanded in direction D if $f_D(n) > W \cdot C^*$.

3.2 W-restrained

Recall that fMM(p) is *restrained* with regards to p. We now extend this property to BSS.

Definition 1 (W-restrained)**.** We say that a BiHS algorithm is *W-restrained* with regards to a fraction $0 \leq p \leq 1$ and a constant $W \geq 1$ if the forward search never expands a node n with $g_F(n) > W \cdot p \cdot C^*$ and the backward search never expands a node n with $g_B(n) > W \cdot (1-p) \cdot C^*$.

When $p = 1/2$ (as in MM), either direction D never expands nodes n with $g_D(n) > W \cdot C^*/2$. By definition, if an algorithm is W-restrained, it is also restrained for every $W' > W$. In particular, MM and fMM are also W-restrained for any $W \geq 1$. Nonetheless, since MM and fMM

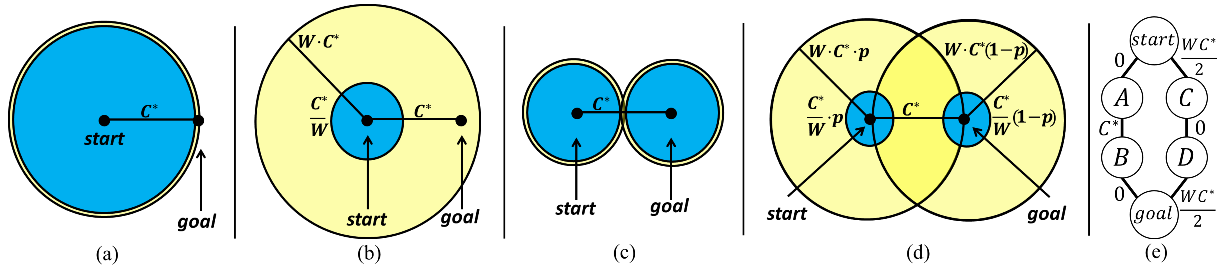

Figure 1: (a) UHS (b) UBSS (c) BiHS (d) BiBSS. Zones: Blue - MEN, Yellow - leeway, White - NEN. (e) Maximal leeway.

are restrained (for $W = 1$) they will always find an optimal solution , thus they cannot utilize the leeway enabled by larger values of W. To this end, we define another property that enables an algorithm to use its entire leeway:

Definition 2 (maximal leeway)**.** A BiBSS algorithm A is said to have a *maximal leeway* if for every weight W and every path P from $start$ to $goal$ of cost $W \cdot C^*$ there exists a problem instance I such that: (1) the optimal path in I from $start$ to $goal$ is of cost C^*, (2) I also contains P, and (3) A can return path P when running on I.

Fig. 1(d) shows the zones for BiBSS. The nodes that are part of MEPs and must be expanded by any W-restrained BiBSS algorithm with $W = 2$ and $p = 1/2$ are depicted as blue circles with radius $C^* \cdot p/W = C^*/4$. The leeway zones are the yellow circles with radius $r = W \cdot C^* \cdot p = C^*$. The *meeting zone* is the intersection of the leeway zones (dark yellow) and is the zone where the frontiers will meet. For $W = 1$ and $p = 1/2$ the meeting zone is exactly the nodes n with $g_D(n) = C^*/2$, hence such algorithms are forced to *meet in the middle*, which is the MM case (Fig. 1(c)).

4 W-restrained Bidirectional Focal Search

Focal Search (FS; also known as A^*_ϵ) (Pearl and Kim 1982; Ebendt and Drechsler 2009; Valenzano et al. 2013) is a UBSS algorithmic scheme that maintains a *focal list*: FOCAL $= \{n \in \text{OPEN} | f(n) \leq W \cdot fmin\}$, where $fmin$ is the lowest f-value in OPEN. Every algorithm that expands *only* nodes from FOCAL is guaranteed to be bounded-suboptimal. FS algorithms vary in the way that they order the focal list.

We next propose a W-restrained *bidirectional focal search* algorithmic scheme (BiFS), which extends FS to BiHS. Given a weight W and a fraction p, BiFS maintains two focal lists $\text{FOCAL}_\text{F} = \{n \in \text{OPEN}_\text{F} | pr_F(n) \leq W \cdot LB\}$ and $\text{FOCAL}_\text{B} = \{n \in \text{OPEN}_\text{B} | pr_B(n) \leq W \cdot LB\}$, where pr_F and pr_B are the priority functions of fMM(p). LB is the lower bound, as defined above for fMM(p). BiFS halts when $U \leq W \cdot LB$, where U maintains the cost of the incumbent solution (as opposed to fMM which halts when $U \leq LB$). Note that just like A* is a special case of fMM (with $p = 0$) where nodes are only chosen from OPEN_F, FS is a special case of BiFS where nodes are only chosen from FOCAL_F.

Implementations of BiFS vary in: (1) how they choose from which side to expand next (FOCAL_F or FOCAL_B) and (2) how they order the nodes in the focal lists (as in FS). Any instantiating of these attributes is W-restrained and is guaranteed to return a bounded-suboptimal solution.

Theorem 2. *BiFS is W-restrained*

Proof. FOCAL_F and FOCAL_B only contain nodes with $pr_F(n) \leq W \cdot LB$ and $pr_B(n) \leq W \cdot LB$. As $pr_F(n) = \max(f_F(n), g_F(n)/p)$ and $pr_B(n) = \max(f_B(n), g_B(n)/(1-p))$, the nodes n in the focal lists have $g_F(n) \leq W \cdot p \cdot LB$ and $g_B(n) \leq W \cdot (1-p) \cdot LB$, respectively. Since $LB \leq C^*$, BiFS will never expand nodes with $g_F(n) > W \cdot p \cdot C^*$ or $g_B(n) > W \cdot (1-p) \cdot C^*$. □

Theorem 3. *BiFS is bounded-suboptimal.*

Proof outline. BiFS terminates either when the open lists are empty or when a solution was found such that $U \leq W \cdot LB$. Since $LB \leq C^*$, BiFS only terminates and returns U if $U \leq W \cdot C^*$. Thus, if a solution is returned, it is guaranteed to be bounded-suboptimal. We now show that BiFS must return a solution, if such exists. Let $P = n_1, \ldots n_k$ be a W-bounded suboptimal solution (of cost $\leq W \cdot C^*$) to the given problem instance. Due to admissibility, for every node n_i and direction D, $g_D(n_i) + h_D(n_i) \leq W \cdot C^*$. Furthermore, for every node n_i, either $g_F(n_i) \leq W \cdot p \cdot C^*$ or $g_B(n_i) \leq W \cdot (1-p) \cdot C^*$. Since either $pr_F(n_i) \leq W \cdot C^*$ or $pr_B(n_i) \leq W \cdot C^*$, all nodes in P will be in the open lists of BiFS. Consequently, BiFS is guaranteed to find a bounded-suboptimal solution, if such exists. □

5 Creating a WA* Version of MM (WMM)

WA* is a special case of FS that does not require to maintain FOCAL in a separate list as all nodes expanded by WA* must be in FOCAL, as was noted by Ebendt and Drechsler (2009). In this section, we develop a WA* version of MM (WMM). That is, a BiBSS algorithm that instantiates BiFS without explicitly maintaining FOCAL lists and, instead, the algorithm orders the OPEN lists similarly to WA*, which guarantees to return a bounded-suboptimal solution. For simplicity, for the remainder of the paper we limit the discussion to the MM case ($p = 1/2$). But, all definitions and proofs can be directly adapted to any fMM(p). In our attempts below, we want to mimic WA* by multiplying the h-value by W.

First Attempt – Note that the priority function of MM can be written as $pr_D(n) = g_D(n) + \max(g_D(n), h_D(n))$ for side D of the search. Therefore, the term $\max(g_D(n), h_D(n))$ can be seen as the h-value in the $f = g + h$ formalism. Thus, the first attempt (denoted WMM1) multiplies that part by W:

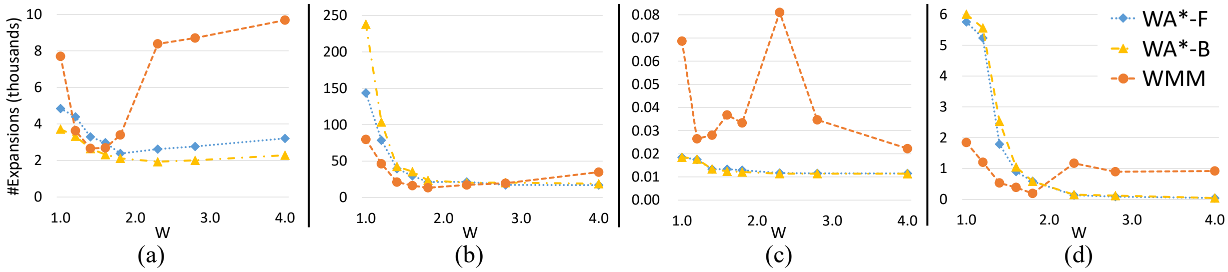

Figure 2: #Expansions on (a) 15-STP with MD (b) 15-STP with MD\5 (c) 14-Pan with GAP (d) 14-Pan with GAP\2

$$prW_D(n) = g_D(n) + W \cdot \max(g_D(n), h_D(n))$$

Clearly, WMM1 is an instantiation of BiFS, and is therefore W-restrained. However, it does not conform to the *maximal leeway* condition defined above (Definition 2). The reason is as follows. Let n be a node with $prW_D(n) \leq W \cdot C^*$. Since $(W + 1) \cdot g_D(n) \leq prW_D(n) \leq W \cdot C^*$, then $g_D(n) \leq \frac{W}{W+1} \cdot C^*$. For $W = 1$, we get $g_D(n) \leq C^*/2$ which is similar to MM. However, for large values of W, $\frac{W}{W+1}$ converges to 1, and $g(n) \leq C^*$. Thus for any value of W, the search will never venture further from $g(n) = C^*$. Given the path $P = start, C, D, goal$ in Figure 1(e), WMM1 will never return P for any problem instance, as $prW_F(C) = prW_B(D) = (1 + W)(W \cdot C^*/2) > W \cdot C^*$.

Second Attempt – The second attempt (denoted WMM2) multiples the h-values by W in a more restrictive way:

$$prW_D(n) = g_D(n) + \max(g_D(n), W \cdot h_D(n))$$

Theorem 4. *WMM2 has a* maximal leeway

Proof outline. Let P be any path of cost $W \cdot C^*$. We can construct I similarly to the problem instance presented in Figure 1(e), except for path P which is given and replaces the path of C and D. Also, we set $h_F(A) = h_B(B) = C^*$, and $\forall n \in P, h_D(n) = 0$. Thus, $prW_F(A) = prW_B(B) = W \cdot C^*$ and $\forall n \in P$ either $prW_F(n) \leq W \cdot C^*$ or $prW_B(n) \leq W \cdot C^*$. Therefore, path P with cost $W \cdot C^*$ can be returned by WMM2. $\square$

In A*, the f-value is monotonically increasing between a node and its children. By contrast, in WA*, since h is inflated by W, a child c can have a smaller $g_D(c) + W \cdot h_D(c)$ than its parent. Thus, in WA* the minimal priority in OPEN can decrease (this might cause reopening of nodes (Sepetnitsky, Felner, and Stern 2016)). The same phenomenon may happen in BiBSS. If the algorithms choose the node with the minimal priority among both sides, it will cause one side of the search to always have the minimal priority (among both sides) which continues to decrease. This phenomenon causes MMW2 to expend the entire leeway in one direction before moving to the other direction, which hinders the task of connecting the frontiers for finding a solution.

Third (final) Attempt – The variant of the third attempt (WMM3) uses the priority function of WMM2, but with a major change. In the first step, WMM3 chooses the side D to expand by alternating between the two sides.[3] Then, it expands a node in the chosen side according to the priority function. This will prevent the starvation of sides described for WMM2. However, to keep WMM3 being W-restrained, if the node n with the lowest priority is outside Focal (i.e., $pr_D(n) > W \cdot LB$), it is not expanded and WMM3 jumps to the other side. Since WMM3 solved both issues of WMM1 and WMM2, we use it for the empirical evaluation. For clarity, hereafter we refer to WMM3 simply as WMM.

[3] Other criteria can be used, such as Pohl's *cardinally criterion*.

6 Experimental Study

It was shown (Holte et al. 2017; Sturtevant et al. 2020) that optimal BiHS algorithms are generally weaker than UHS algorithms with an accurate heuristic, but outperform them with mediocre or weak heuristics. Our aim here is not to develop the strongest BiBSS algorithm possible, but rather to explore whether the same trend is valid for BiBSS.

We experimented on the 15-sliding tile puzzle (15-STP) and the 14-pancake puzzle (14-Pan) with WA* and WMM (Figure 2) with W values that range from 1 to 4 (x-axis). We report the number of expansions (y-axis) averaged over 100 random instances. As some domains are unbalanced, we executed forward- (WA*-F) and backward WA* (WA*-B).

Figure 2(a) presents the results for 15-STP with the Manhattan Distance heuristic (MD). Here, WA* outperformed WMM. When we weakened the heuristic to MD\5 (tiles 1 to 5 were not considered), Figure 2(b) shows that WMM outperformed WA* for $W \leq 2.5$. Figures 2(c) and 2(d) show the results on 14-Pan with the GAP heuristic and with the less accurate GAP\2 heuristic (which excludes the two smallest pancakes). In both domains we see the same known trend that, with a weak heuristic, BiBSS outperforms its UHS counterpart. Nonetheless, we observe a clear transition in performance at around $W = 2$. For $W \geq 2$ *goal* is part of the leeway zone of the forward search and vice versa. Deeper study is needed to completely understand this impact.

7 Conclusion and Future Work

This paper takes a first step towards extending recent theoretical advancements in BiHS to BSS (BiBSS), and introduces WMM which is based on two fundamental algorithms, WA* and MM. Future work will develop more BiBSS algorithms by hybridizing more advanced UBSS and BiHS algorithms. WMM can also be enhanced by dividing W unevenly between the two sides of the search, e.g., by setting W_F and W_B such that $W = W_F \cdot p + W_B \cdot (1 - p)$.

Acknowledgements

This work was funded by the United States-Israel Binational Science Foundation (BSF) under grant numbers 2017692 and 2021643, and by the Canada CIFAR AI Chairs Program. We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

Alcázar, V. 2021. The Consistent Case in Bidirectional Search and a Bucket-to-Bucket Algorithm as a Middle Ground between Front-to-End and Front-to-Front. In *ICAPS*, 7–15.

Alcázar, V.; Riddle, P. J.; and Barley, M. 2020. A Unifying View on Individual Bounds and Heuristic Inaccuracies in Bidirectional Search. In *AAAI*, 2327–2334.

Barker, J. K.; and Korf, R. E. 2015. Limitations of front-to-end bidirectional heuristic search. In *AAAI*, 1086–1092.

Chen, J.; Holte, R. C.; Zilles, S.; and Sturtevant, N. R. 2017. Front-to-End Bidirectional Heuristic Search with Near-Optimal Node Expansions. In *IJCAI*, 489–495.

Chen, J.; and Sturtevant, N. R. 2019. Conditions for Avoiding Node Re-Expansions in Bounded Suboptimal Search. In *IJCAI*, 1220–1226.

Chen, J.; and Sturtevant, N. R. 2021. Necessary and Sufficient Conditions for Avoiding Reopenings in Best First Suboptimal Search with General Bounding Functions. In *AAAI*, 3688–3696.

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

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

Eckerle, J.; Chen, J.; Sturtevant, N. R.; Zilles, S.; and Holte, R. C. 2017. Sufficient conditions for node expansion in bidirectional heuristic search. In *ICAPS*, 79–87.

Fickert, M.; Gu, T.; and Ruml, W. 2022. New Results in Bounded-Suboptimal Search. In *AAAI*, 10166–10173.

Gilon, D.; Felner, A.; and Stern, R. 2016. Dynamic Potential Search—A New Bounded Suboptimal Search. In *SoCS*, 36–44.

Hart, P. E.; Nilsson, N. J.; and Raphael, B. 1968. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2): 100–107.

Holte, R. C.; Felner, A.; Sharon, G.; Sturtevant, N. R.; and Chen, J. 2017. MM: A bidirectional search algorithm that is guaranteed to meet in the middle. *Artificial Intelligence*, 252: 232–266.

Islam, F.; Narayanan, V.; and Likhachev, M. 2016. A-Connect: Bounded suboptimal bidirectional heuristic search. In *ICRA*, 2752–2758.

Kwa, J. B. H. 1989. BS*: An admissible bidirectional staged heuristic search algorithm. *Artificial Intelligence*, 38(1): 95–109.

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

Pearl, J.; and Kim, J. H. 1982. Studies in Semi-Admissible Heuristics. *IEEE Trans. Pattern Anal. Mach. Intell.*, 4(4): 392–399.

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

Pohl, I. 1971. Bi-directional search. *Machine Intelligence*, 6: 127–140.

Sepetnitsky, V.; Felner, A.; and Stern, R. 2016. Repair Policies for Not Reopening Nodes in Different Search Settings. In *SoCS*, 81–88.

Shaham, E.; Felner, A.; Chen, J.; and Sturtevant, N. R. 2017. The Minimal Set of States that Must Be Expanded in a Front-to-End Bidirectional Search. In *SoCS*, 82–90.

Shaham, E.; Felner, A.; Sturtevant, N. R.; and Rosenschein, J. S. 2018. Minimizing Node Expansions in Bidirectional Search with Consistent Heuristics. In *SoCS*, 81–98.

Shperberg, S.; Felner, A.; Shimony, S.; Sturtevant, N.; and Hayoun, A. 2019a. Improving bidirectional heuristic search by bounds propagation. In *SoCS*, 106–114.

Shperberg, S. S.; Danishevski, S.; Felner, A.; and Sturtevant, N. R. 2021. Iterative-deepening Bidirectional Heuristic Search with Restricted Memory. In *ICAPS*, 331–339.

Shperberg, S. S.; Felner, A.; Sturtevant, N. R.; Shimony, S. E.; and Hayoun, A. 2019b. Enriching non-parametric bidirectional search algorithms. In *AAAI*, 2379–2386.

Sturtevant, N. R.; Shperberg, S. S.; Felner, A.; and Chen, J. 2020. Predicting the Effectiveness of Bidirectional Heuristic Search. In *AAAI*, 281–290.

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

Valenzano, R. A.; Arfaee, S. J.; Thayer, J. T.; Stern, R.; and Sturtevant, N. R. 2013. Using Alternative Suboptimality Bounds in Heuristic Search. In *ICAPS*, 233–241.