

On the Differences and Similarities of fMM and GBFHS

Shahaf S. Shperberg

Computer Science Dept.
Ben-Gurion University
Be'er-Sheva, Israel
shperbsh@post.bgu.ac.il

Ariel Felner

SISE Dept.
Ben-Gurion University
Be'er-Sheva, Israel
felner@bgu.ac.il

Abstract

fMM and GBFHS are two prominent parametric bidirectional heuristic search algorithms. A great deal of theoretical and empirical work has been done on both of these algorithms over the past few years. A number of interesting theoretical properties were proved for only one of these algorithms. In this paper we analyze the differences and similarities between these algorithms by comparing their minimal number of node expansions, and their implementations. Importantly, we introduce a version of fMM, called dfMM, that uses a dynamic fraction, and show that when both algorithms are enriched by lower-bound propagation they become equivalent. In particular, for every parameter value of dfMM_{lb} we provide a parameter value of GBFHS_{lb} such that both algorithms expand the same sequence of nodes, and vice versa. This equivalence indicates that all theoretical properties proved for one algorithm hold for both. Therefore, it suffices to consider only one of these algorithms for future analyses and benchmarks.

1 Introduction

Over the past few years, bidirectional heuristic search (Bi-HS) has been the focus of an active research that yielded new algorithms and many theoretical and empirical results. Notable examples for such algorithms are *fMM* (Shaham et al. 2017) and *GBFHS* (Barley et al. 2018), two *parametric* Bi-HS algorithms. On the surface, these algorithms seem very different, but this paper shows their similarities: they both have the ability to set the meeting point between the two frontiers using a user-supplied parameter. To possess this ability, fMM prioritizes nodes by dividing their g -values by a parameterized *fraction*, while GBFHS splits the current solution bound using a parameterized *split function* that limits which nodes are expanded from each side. In this paper we discuss the differences and similarities of the algorithms in terms of minimal node expansions. In addition, we show that both GBFHS and fMM can be enriched by the *lower-bound propagation* (*lb*-propagation) (Shperberg et al. 2019b), creating fMM_{lb} and GBFHS_{lb} respectively. We show that adding *lb*-propagation to both algorithms never harm their performance (in terms of minimal node expansions required to solve and verify optimal solutions) and the performance even improves for many problem instances.

Copyright © 2020, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Moreover, we define a variant of fMM, called dfMM, that uses a dynamic fraction (that changes during the search) instead of fixed fraction. dfMM can also be enriched by the *lb*-propagation to create dfMM_{lb}. Next, we show that dfMM_{lb} and GBFHS_{lb} are *equivalent*. This means that every fraction used by dfMM_{lb} can be mapped to a split function of GBFHS_{lb}, and vice versa, such that both algorithms expand the same sequence of nodes (i.e. the same set of nodes in the same order). dfMM_{lb} and GBFHS_{lb} retain the theoretical properties of both original algorithms, with the addition of other desirable properties induced by the *lb*-propagation. Thus, it suffices to consider only one of these algorithms for future analyses and benchmarks.

Finally, we examine implementations of all algorithms (fMM, dfMM, fMM_{lb}, dfMM_{lb}, GBFHS, and GBFHS_{lb}) and compare their complexity. We show that fMM with a fixed fraction imposes the least amount of overhead per node expansion, while all the rest algorithms require similar amount of overhead.

2 Definitions and Background

Our aim is to find the the least-cost path between *start* and *goal* in a given implicit graph G . Let $d(x, y)$ denote the shortest distance between x and y and let $C^* = d(\text{start}, \text{goal})$. In some cases, the cost of the cheapest edge of the graph (denoted by ϵ) is available. Otherwise ϵ is assumed to be 0.

Bi-HS typically keeps two open lists, $Open_F$ for the forward search (F), and $Open_B$ for the backward search (B). Both fMM and GBFHS use *front-to-end* heuristics (Kaindl and Kainz 1997) which estimate the distance between any state and the *start* or *goal*. *Front-to-front* heuristics (de Champeaux and Sint 1977) estimate the distance between any two nodes in the search space and are beyond our scope.

Given a direction D (either F or B), we use f_D , g_D and h_D to indicate f -, g -, and h -values in direction D . In addition, f_{min_D} and g_{min_D} represent the minimal f - and g -values in that direction. We use \bar{D} to denote the direction opposite to D , and define $f_{\bar{D}}$, $g_{\bar{D}}$ and $h_{\bar{D}}$ symmetrically.

The *forward heuristic* h_F is *admissible* if and only if $h_F(u) \leq d(u, \text{goal})$ for every state $u \in G$ and is *consistent* if and only if $h_F(u) \leq d(u, u') + h_F(u')$ for all $u, u' \in G$. Properties of the *backward heuristic* h_B are defined analogously. Let I_{AD} be the set of problems with ad-

missible heuristics, and $I_{\text{CON}} \subseteq I_{\text{AD}}$ be the set of problems with consistent heuristics. A search algorithm is *admissible* on a set of problem instances I if it is guaranteed to find an optimal solution on every problem instance $i \in I$.

2.1 Guaranteeing Solution Optimality

Dechter and Pearl (1985) showed that any *unidirectional* search algorithm¹ that is admissible on I_{AD} , must expand all nodes n with $f(n) < C^*$ in order to guarantee optimality of solutions, when given a problem instance from I_{CON} .

Eckerle et al. (2017) generalized this to Bi-HS by examining pairs of nodes $\langle u, v \rangle$ such that $u \in \text{Open}_F$ and $v \in \text{Open}_B$. If u and v meet all the following conditions:²

1. $f_F(u) < C^*$
2. $f_B(v) < C^*$
3. $g_F(u) + g_B(v) + \epsilon < C^*$

then every any Bi-HS algorithm¹ that is admissible on I_{AD} must expand at least one of u or v to ensure that there is no path from *start* to *goal* passing through u and v of cost $< C^*$, when given a problem instance from I_{CON} . Such a pair of nodes is called a *must-expand pair* (MEP). These conditions can be formalized into a pairwise lower-bound:

Definition 1. For each pair of nodes (u, v) let $lb(u, v) = \max\{f_F(u), f_B(v), g_F(u) + g_B(v) + \epsilon\}$

As a result $\langle u, v \rangle$ is a *must-expand pair* if $lb(u, v) < C^*$.

A* expands all the nodes with $f_F < C^*$, which is equivalent to expanding the forward node of every MEP. Bi-HS algorithms may expand nodes from both sides, potentially covering all the MEPS with fewer expansions.

Shperberg et al. (2019b) extended the lb notation and defined a lower-bound for nodes instead of pairs. In order to define a lower-bound to a node u , the bound $lb(u, v)$ is applied to every node v on the opposite frontier of u , and only the minimum among these values is considered. Formally,

$$lb(u) = \min_{v \in \text{open}_B} lb(u, v)$$

$lb(u)$ is a lower-bound on the cost of any solution that passes through u for every node u in Open_D .

We note that different settings, in which algorithms that are admissible on I_{CON} but not on I_{AD} , have been explored (Shaham et al. 2018; Alcazar, Riddle, and Barley 2020). In these settings, the conditions of Eckerle et al. (2017) are too strict, i.e., not every MEP (as defined above) needs to be covered, and a different set of conditions is required to define MEPS and bound the minimal number of node expansions for these cases. In this paper we consider GBFHS and fMM, which are both admissible on I_{AD} . Thus, these other settings are beyond the scope of this paper.

2.2 Well-Behavedness and Reasonableness

Shperberg et al. (2019b) defined two desirable properties for Bi-HS algorithms the *well-behavedness* property and the

¹Only DXBB algorithms (Eckerle et al. 2017) are considered here. These are deterministic algorithms that are expansion-based (have only black-box access to the graph G via expansion methods)

²The ϵ term was added later by Shaham et al. (2018) as a generalization of the inequalities.

reasonableness property. Let $\mathcal{A}_h(I, t)$ be the sequence of nodes expanded by running algorithm \mathcal{A} using heuristic h on problem instance I with a tie-breaking function t , and let $S(\mathcal{A}_h(I, t))$ be a (unordered) set of nodes induced by the expansion performed by $\mathcal{A}_h(I, t)$.

Definition 2. Let h_1, h_2 be admissible, consistent heuristics, such that h_1 dominates h_2 . Algorithm \mathcal{A} is said to be well-behaved if for every tie-breaking policy t and problem instance I , there exists a tie-breaking policy t' such that $S(\mathcal{A}_{h_1}(I, t')) \subseteq S(\mathcal{A}_{h_2}(I, t))$.

Definition 3. A Bi-HS algorithm is reasonable if for every tie-breaking policy it does not expand a node v if either $lb(v) > C^*$, or if $lb(v) = C^*$ and a solution of cost C^* was found.

In some sense, the *well-behavedness* property ensures that improving the heuristic function never harms the algorithms performance (up to a tie-breaking). By contrast, the *reasonableness* property ensures that algorithms do not expand nodes whose lower-bound cannot lead to optimal solutions.

2.3 Fractional MM

MM is a Bi-HS algorithm, which is admissible on I_{AD} , that *meets in the middle* (Holte et al. 2017). I.e. it is guaranteed to never expand a node whose g -value exceeds $C^*/2$. Fractional MM (fMM(p)) is a generalization of MM that never expands nodes in the forward direction whose g -value exceeds $p \cdot C^*$, and never expands nodes in the backward direction whose g -value exceeds $(1 - p) \cdot C^*$. For a given fraction $0 < p < 1$, fMM(p) chooses a node for expansion according to the following priority functions:³

$$\begin{aligned} pr_F(u) &= \max\{g_F(u) + h_F(u), \frac{g_F(u)}{p} + \epsilon\} \\ pr_B(v) &= \max\{g_B(v) + h_B(v), \frac{g_B(v)}{1-p} + \epsilon\} \end{aligned}$$

The node with a lowest priority is chosen for expansion regardless of whether it is in the forward or backward side (ties can be broken in many ways). Note that MM is a special case of fMM(p) with $p = 1/2$.

The priority of a node n in direction D can be written as:

$$pr_D(n) = \max\{f_D(n), \frac{g_D(n)}{q_D} + \epsilon\}$$

where $q_F = p$ and $q_B = (1 - p)$.

fMM terminates when one of these conditions holds:

- (1) One of Open_F or Open_B is empty.
- (2) There exists a node v in both open lists with $C = g_F(v) + g_B(v)$ such that either:
 - (i) $f_{\min_F} \geq C$;
 - (ii) $f_{\min_B} \geq C$;
 - (iii) $g_{\min_F} + g_{\min_B} + \epsilon \geq C$; or
 - (iv) $\min_{u \in \text{Open}_D} pr_D(u) \geq C$.

³Strictly speaking for $p = 1$ or $p = 0$ fMM should run forward- or backward A*. Additionally, the original definition of fMM and MM did not include ϵ , which was introduced in later versions of the algorithms: MM ϵ and fMM ϵ .

Shaham et al. (2017) showed that for every problem instance, there exists a fraction p^* such that $\text{fMM}(p^*)$ is *optimally efficient* and will expand the minimal number of nodes required to guarantee the optimality of its solution, with respect to the setting described in Section 2.1. However, p^* is not known a priori since it depends on search tree structure and the value of C^* . Finally, fMM was shown to be neither reasonable nor well-behaved (Shperberg et al. 2019b).

Example 1. Consider the problem instance that previously appeared in (Holte et al. 2017; Shperberg et al. 2019b) and is depicted in Figure 1. In this problem instance $\epsilon = 1$, and the values inside nodes are h -values in the direction indicated by the arrow. fMM that meets in the middle (with $p = 1/2$) starts by expanding *start* and *goal* (priority of 2 due to their f -value), after which nodes A , C , S_1 and G_1 are generated. Since the f -value of all these nodes is *greater than or equal to* twice their g -values + ϵ , their priority is determined by their f -value. Therefore, S_1 and G_1 have a priority of 3, while A and C have a priority of 4. Thus, for any tie-breaking, fMM must expand S_1 and G_1 before expanding A and C which are required in order to find a solution and terminate.

2.4 GBFHS

GBFHS (general breadth-first heuristic search) (Barley et al. 2018) is a prominent bidirectional heuristic search algorithm that iteratively increases the depth of the search. For each depth, denoted by $fLim$, GBFHS uses a pre-defined *split function* (a "parameter" of the algorithm) that determines how deep to search on each side. The split function splits $fLim$ to $gLim_F$ and $gLim_B$, such that $fLim = gLim_F + gLim_B + \epsilon - 1$ (in unit edge cost domains $\epsilon - 1 = 0$). For a given iteration (i.e., a given value of $fLim$) all nodes with $f_D(n) \leq fLim$ and $g_D(n) < gLim_D$ are called *expandable*. GBFHS expands all *expandable* nodes from both directions. GBFHS terminates as soon as a solution with $cost = fLim$ is found. Specifically, GBFHS stops when there exists a node n in both open lists with $g_F(n) + g_B(n) \leq fLim$. If a solution is not found after expanding all *expandable* nodes, $fLim$ is incremented (adds 1), and as a result the split function updates $gLim_F$ or $gLim_B$ (such that $fLim = gLim_F + gLim_B + \epsilon - 1$). Then, a new iteration begins. The split function must update the g -limits ($gLim_F$ and $gLim_B$) in a consistent way, i.e., the values it returns must be *larger than or equal to* the previous values. This means that when $fLim$ is incremented then one of $gLim_F$ and $gLim_B$ is incremented too.

GBFHS was shown to have some desirable properties when given a problem instance from I_{AD} : **(1)** it returns an optimal solution when the edges are non-negative integers; **(2)** in unit cost domains, the first solution GBFHS finds is guaranteed to be optimal; **(3)** Its frontiers can be made to meet anywhere using a proper split function; and **(4)** it is both *well-behaved* and *reasonable*.

Since GBFHS is guaranteed to return an optimal solution only when the edge costs are (non-negative) integers, we will assume such edge costs for the sake of the analysis. Nevertheless, we conjecture that GBFHS can be slightly modified

in a way that would guarantee optimal solutions for any non-negative edge costs, while retaining all of its original properties. Investigating this conjecture is left for future work.

Example 2. Consider Figure 1 again. GBFHS that meets in the middle will then split $fLim$ to $gLim_F = gLim_B = 1$. Thus, GBFHS expands every node with $f_D \leq 2$ and $g_D < 1$. Only *start* and *goal* meet these conditions and they are expanded and as a result, A, C, S_1 and G_1 are generated. Since there are no expandable nodes left, $fLim$ is increased to 3, and the new split function is either ($gLim_F = 2$ and $gLim_B = 1$) or ($gLim_B = 1$ and $gLim_F = 2$). Since the problem graph is symmetric, without loss of generality we assume that GBFHS chooses the split $gLim_F = 2$ and $gLim_B = 1$. Thus, only the S_1 nodes are expanded, and then $fLim$ is increased to 4. Now, $gLim_F = 2$ and $gLim_B = 2$, and the set of expandable nodes is $\{A, C, G_1\}$. Therefore, GBFHS can expand A and C , and terminate without having to expand G_1 .

2.5 The lb -propagation

Shperberg et al. (2019b) proposed a way for enhancing heuristics by propagating lower-bounds (lb -propagation) between frontiers. The lb -propagation is based on the *must-expand theory* of Eckerle et al. (2017) (Section 2.1), by bounding the minimal solution cost that can pass through each node u in the open lists.

Given a heuristic function h_D , for nodes in direction D , they defined $h_{D_{lb}}(n) = lb(n) - g_D(n)$ to be the new heuristic value results by the lb -propagation. Since $lb(n) \geq f_D(n)$, $h_{D_{lb}}(n) \geq h_D(n)$ for every node. Thus this new heuristic is at least as strong as the original heuristic. In addition, the f -value of a node n using $h_{D_{lb}}$, denoted by $f_{D_{lb}}(n)$ ($= g_D(n) + h_{D_{lb}}(n)$), equals $lb(n)$.

The priority function of fMM_{lb} is:

$$pr_D(n) = \max\{f_{D_{lb}}(n), \frac{g_D(n)}{q_D} + \epsilon\}$$

In addition, since lb -propagation already incorporates the information of f - and g - values from nodes in the other direction into the priority function, the first three inequalities in the stopping criteria above in Section 2.3 become redundant. Thus, fMM_{lb} halts when

$$\min_{u \in \text{Open}_D} pr_D(u) \geq C$$

This lb -propagation can bestow some desirable properties (e.g. being *reasonable* and *well-behaved*) on some existing algorithms. In particular, adding lb -propagation makes fMM_{lb} reasonable and well-behaved (Shperberg et al. 2019b). GBFHS is already reasonable and well-behaved in its original form. However, the lb -propagation may also be applied to GBFHS in a manner that further improves its performance, an issue we further examine below.

3 Dynamic Fractional MM

Recall that GBFHS is defined using a split function which returns a new $gLim_F$ and $gLim_B$ values every time $fLim$ is increased. By contrast, fMM is defined with a fixed fraction

p . We now introduce dfMM, a variant of fMM that uses a dynamic fraction.

dfMM uses the same priority function and termination conditions of fMM. However, the fraction p is updated every time that the minimal pr (among both sides) is changed. We restrict the fraction updates to be consistent with nodes that have already been expanded, i.e. the priority of nodes that have already been expanded must be *less than or equal to* the new minimal priority in the open list. This resembles the restriction of the split function of GBFHS, in which the $gLim_D$ values it returns must be *larger than or equal to* the old $gLim_D$ values. Formally, let p and pr be the fraction and minimal priority (respectively) before the update, and let p' and pr' be the fraction and minimal priority after the update. Since pr was the minimal priority before the fraction update, every node n with $f_D(n) \leq pr$ and $\frac{g_D(n)}{q_D} + \epsilon \leq pr$ has been expanded. We need to make sure that these nodes must have been also expanded using the new fraction, hence, $\frac{g_D(n)}{q_D} + \epsilon \leq pr'$ (where $q'_F = p$ and $q'_B = (1 - p)$). Therefore, we need to make sure that $q_D \cdot (pr - \epsilon) \leq q'_D \cdot (pr' - \epsilon)$. This gives us the following two inequalities:

$$p \cdot (pr - \epsilon) \leq p' \cdot (pr' - \epsilon)$$

and

$$(1 - p) \cdot (pr - \epsilon) \leq (1 - p') \cdot (pr' - \epsilon)$$

which together bound p' as follows:

$$\frac{p \cdot (pr - \epsilon)}{(pr' - \epsilon)} \leq p' \leq 1 - \frac{(1 - p) \cdot (pr - \epsilon)}{(pr' - \epsilon)}$$

By always keeping track of the current and previous minimal priorities in the open lists, as well as the previous fraction, validation of the constraints on p' can be done very efficiently. However, using a new fraction has implementation repercussions, as the open list might need to be reordered every time the fraction changes. This is further discussed in Section 6. Nevertheless, nodes are always expanded according to the most recent priority function.

Example 3. In Figure 1, Similar to fMM, dfMM with $p = 1/2$ starts by expanding *start* and *goal* (priority of 2 due to their f -value), after which nodes A , C , S_1 and G_1 are generated. Now, the minimal priority of nodes in OPEN has been increased and is equal to 3 (as shown in Example 1). Therefore, the fraction can be updated. For example, $p = 2/3$ can be chosen as the new fraction, as it meet the consistency constraint. With this fraction, as with $p = 1/2$, S_1 and G_1 have a priority of 3, while A and C have a priority of 4. Thus, S_1 and G_1 must be expanded before the minimal priority is updated. Finally, p can be updated and afterwards A and C can be expanded (depending on the new fraction) before dfMM terminates.

Clearly, every fixed fraction can be represented by a dynamic fraction (by maintaining the same fraction throughout the search). Moreover, for every dfMM with dynamic fraction there exists an equivalent fMM with fixed fraction, using the value of the dynamic fraction at the end of the search, as a fixed value. Therefore, the theoretical properties of fMM still apply for dfMM. Despite having the same

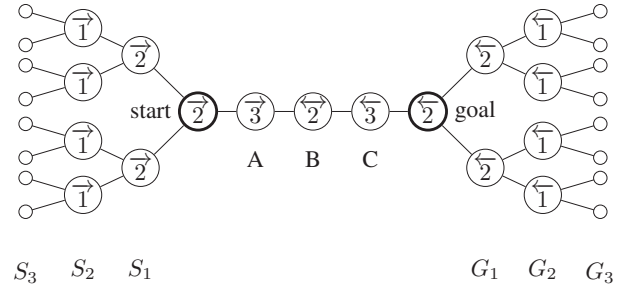


Figure 1: Example where $MNE_{GBFHS}(I) < MNE_{fMM}(I)$

theoretical properties, dynamic fractions allow the usage of information which is only available during the search to decide which nodes are expanded.

dfMM can also be enriched by the lb -propagation, resulting in $dfMM_{lb}$. Similarly to fMM and dfMM every fixed fraction of fMM_{lb} can be represented by a dynamic fraction of $dfMM_{lb}$. By contrast, here it is unclear if for every $dfMM_{lb}$ there exists an equivalent fMM_{lb} . While we conjecture that there exists such an equivalence, this claim is much more challenging to prove (or disprove), since the heuristic values depend on the history of node expansions when using the lb -propagation. However, the theoretical properties of fMM_{lb} still apply for $dfMM_{lb}$. Due to the fraction consistency constraints, dfMM and $dfMM_{lb}$ never expand nodes whose g -value exceed $q_D \cdot C^*$, where q_D is the previous fraction. Thus, both dfMM and $dfMM_{lb}$ are still optimally efficient (with respect to the assumptions mentioned in Section 2.1). In addition, $dfMM_{lb}$ maintains the well-behavedness and reasonableness properties, as Shperberg et al. (2019b) original proofs for fMM_{lb} are not affected by the fact that the fraction is now dynamic.

4 Minimal Expansions of GBFHS and fMM

In order to compare the variants of fMM, and GBFHS, we would like to reason about the number of nodes expanded by each of them regardless of which tie-breaking policy is used. Therefore, we use $MNE_A(I)$ to denote the Minimal Number of Expansions required by algorithm A to find an optimal solution and guarantee its optimality when running on instance I . Formally, $MNE_A(I)$ is the number of expansions achieved by running A using the best possible tie-breaking, which is not known a priori.⁴

Both fMM and GBFHS are parametric algorithms. For simplicity, we will consider fMM and GBFHS that aim at meeting in the middle. However, the analysis below can be easily modified to any meeting point. We begin by considering the basic variants, namely, GBFHS and fMM.

Theorem 1. *There exists a problem instance I for which $MNE_{GBFHS}(I) < MNE_{fMM}(I)$*

Proof. Recall the execution of fMM and GBFHS on the problem instance I in Figure 1, described in Example 1 and

⁴A somewhat less formal definition of this notion appeared in Barley et al. (2018) under the name *lower-bound*.

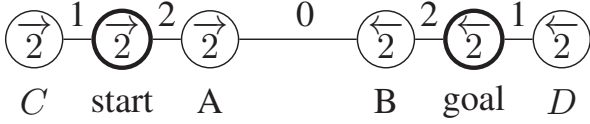


Figure 2: Example in where $MNE_{fMM_{lb}}(I) < MNE_{GBFHS}(I)$

Example 2 respectively. fMM has to expand *start*, *goal*, $2 \cdot S_1$, $2 \cdot G_1$, *A* and *C* before been able to terminate, thus, $MNE_{fMM}(I) = 8$. By contrast, GBFHS does not need to expand both S_1 and G_1 before finding a solution. Therefore, $MNE_{GBFHS}(I) = 6$ (*start*, *goal*, $2 \cdot (S_1$ or $G_1)$, *A* and *C*). Note that the number of nodes in G_1 and S_1 can be arbitrarily large. Therefore, the difference between $MNE_{GBFHS}(I)$ and $MNE_{fMM}(I)$ cannot be bounded by any constant. \square

Theorem 2. For every problem instance I , $MNE_{GBFHS}(I) \leq MNE_{fMM}(I)$

Proof. Assume in contradiction that there exists a problem instance I for which $MNE_{GBFHS}(I) > MNE_{fMM}(I)$, and let n be the first node in direction D that is expanded by GBFHS and not by fMM given the same tie breaking policy for both algorithms. Since n is expanded by GBFHS, we know that $f_D(n) \leq fLim$, in addition, $g_D(n) < gLim_D$. We consider GBFHS that meets in the middle. Therefore, $gLim_D \leq \lceil \frac{fLim - \epsilon + 1}{2} \rceil$.

Case 1: $\frac{fLim - \epsilon + 1}{2}$ is integer. In this case, $g_D(n) < \frac{fLim - \epsilon + 1}{2}$. Since we assume integer costs, $g_D(n) \leq \frac{fLim - \epsilon + 1}{2} - 1$. By multiplying both sides by 2 and moving ϵ to the left hand side in the last inequality, we get that $\frac{g_D(n)}{1/2} + \epsilon \leq fLim + 1 - 2 = fLim - 1$.

Case 2: $\frac{fLim - \epsilon + 1}{2}$ is not integer. In this case, $g_D(n) < \frac{fLim - \epsilon + 1}{2} + 1/2$. Using the integer costs assumption, $g_D(n) \leq \frac{fLim - \epsilon + 1}{2} - 1/2$. Thus, $\frac{g_D(n)}{1/2} + \epsilon \leq fLim - \epsilon + 1 - 1 + \epsilon = fLim$.

In both cases $pr_D(n) = \frac{g_D(n)}{1/2} + \epsilon \leq fLim$. We assumed that fMM found a solution without expanding n , thus, it found a solution via a different node n' , such that $pr_D(n') \leq pr_D(n) \leq fLim$. Since fMM chose n' instead of n , we know that $f_D(n') \leq fLim$ and $\frac{g_D(n')}{1/2} + \epsilon \leq fLim$, and therefore, $g_D(n') \leq \frac{fLim - \epsilon}{2} < gLim_D$. Thus, both n' and n are expandable by GBFHS, and since fMM and GBFHS use the same tie-breaking policy, GBFHS had to expand n' and not n . \square

It follows from the previous two theorems that GBFHS dominates fMM with respect to the MNE measure. In essence, for every problem instance the MNE value of GBFHS is bounded by the MNE value of fMM, and there exist cases in which the MNE value of GBFHS is strictly

smaller. We will now prove that the dynamic version fMM enriched by *lb*-propagation ($dfMM_{lb}$) dominates GBFHS.

We begin by showing that there exist cases in which the MNE value of fMM_{lb} (with static fraction) is strictly smaller than the MNE value of GBFHS.

Theorem 3. There exists a problem instance I for which $MNE_{fMM_{lb}}(I) < MNE_{GBFHS}(I)$

Proof. Consider the execution of GBFHS that meets in the middle on the problem instance I in Figure 2, in which $\epsilon = 0$ and the values inside nodes are h -values in the direction indicated by the arrow. At the beginning of the search, $fLim = 2$, the f -value of *start* and *goal*. A GBFHS that meets in the middle will then split $fLim + 1 - \epsilon$ to either ($gLim_F = 2$ and $gLim_B = 1$) or ($gLim_B = 1$ and $gLim_F = 2$). Since the problem graph is symmetric, we can assume that GBFHS chose the split $gLim_F = 2$ and $gLim_B = 1$ without loss of generality. Thus, GBFHS expands every node with $f_F \leq 2$ and $g_F < 2$, and every node with $f_B \leq 2$ and $g_B < 1$. The only nodes that meet these conditions are *start* and *goal*, they are expanded and as a result *A*, *B*, *C* and *D* are generated. Since all expandable nodes have been expanded, $fLim$ is increased to 3, and the new split function is $gLim_F = 2$ and $gLim_B = 2$. Now, every node with $f_D \leq 3$ and $g_D < 2$ is expandable. Therefore, GBFHS expands *C* and *D* before increasing $fLim$ and before expanding *A* or *B* in order to find a guaranteed optimal solution and terminate. Thus, $MNE_{GBFHS}(I) = 5$ (*start*, *goal*, *C*, *D*, and (*A* or *B*)).

In contrast, fMM_{lb} (with fraction $p = 1/2$) starts by expanding either *start* or *goal* (priority of 2). Without loss of generality, fMM_{lb} expands *start*, after which $f_{B_{lb}}(goal) = \min\{lb(C, goal) = 3$. Thus, *goal* and *C* have a priority of 3, while the priority of *A* is equal to 4. fMM_{lb} can then choose to expand *goal*; afterwards, since $lb(C) = lb(C, D) = 3$, fMM_{lb} can then expand *C*. After expanding *C*, $lb(A, D) = lb(A, B) = 4$, thus, the priority of all nodes in OPEN is 4, and fMM_{lb} can choose to expand *A* in order to find a guaranteed optimal solution and terminate. Hence, $MNE_{fMM_{lb}}(I) = 4$ (*start*, *goal*, *C*, and (*A* or *B*)). Note that there can be many nodes similar to *D*, therefore, the difference between $MNE_{fMM_{lb}}(I)$ and $MNE_{GBFHS}(I)$ also cannot be bounded by any constant. \square

Since every fixed fraction can be represented by a dynamic fraction, $MNE_{dfMM_{lb}}(I) < MNE_{GBFHS}(I)$ for the same problem instance in the proof of Theorem 3.

Note that since the heuristic modified by the *lb*-propagation is at least as strong as the original heuristic, $f_{D_{lb}}(n) \geq f_D(n)$ for every node n . In addition, GBFHS is well-behaved, and therefore using a better heuristic never harm its performance. Thus, $MNE_{GBFHS_{lb}}(I) \leq MNE_{GBFHS}(I)$. Following the last inequality, along with the fact that $MNE_{dfMM_{lb}}(I) = MNE_{GBFHS_{lb}}(I)$ (as we prove in the next section), shows that $MNE_{dfMM_{lb}}(I) \leq MNE_{GBFHS}(I)$. Since there exists a problem instance I for which $MNE_{dfMM_{lb}}(I) < MNE_{GBFHS}(I)$ (Theorem 3) and $MNE_{dfMM_{lb}}(I) \leq MNE_{GBFHS}(I)$, $dfMM_{lb}$ dominates GBFHS with respect to the MNE measure. Consequently,

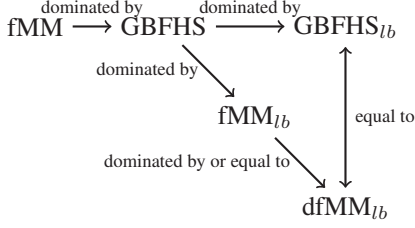


Figure 3: MNE hierarchy for all variants

since $\text{MNE}_{\text{dfMM}_{lb}}(I) = \text{MNE}_{\text{GBFHS}_{lb}}(I)$, GBFHS_{lb} also dominates GBFHS . Figure 3 summarize the MNE hierarchy of the fMM and GBFHS variants.

5 dfMM_{lb} is Equivalent to GBFHS_{lb}

In this section we show the equivalence of dfMM_{lb} and GBFHS_{lb} . In essence, we show that every fraction of dfMM_{lb} can be mapped to a split function of GBFHS_{lb} , and vice versa, such that both algorithms expand the same sequence of nodes (the same set of nodes in the same order).

To show the equivalence, we need to: (1) define appropriate split function for GBFHS_{lb} given a fraction p of dfMM_{lb} ; and (2) define an appropriate dynamic fraction p for dfMM_{lb} given a split $\langle g\text{Lim}_F, g\text{Lim}_B \rangle$ for every value of $f\text{Lim}$. We will tackle both tasks in their respective order.

To utilize the benefits of the lb -propagation, we require a tie-breaking that favors nodes from one side (either forward or backward) (Shperberg et al. 2019b). For fMM_{lb} this means that all nodes in the favored direction are expanded before nodes in the opposite direction with the same priority, whose priority is induced by their g -value. Formally, let D be the favored direction. Before expanding node n' in direction \bar{D} with priority $pr_{\bar{D}}(n') = \frac{g_{\bar{D}}(n')}{q_{\bar{D}}} + \epsilon$, every node n with in direction D with priority $pr_D(n) = pr_{\bar{D}}(n')$ has already been expanded. For GBFHS_{lb} all nodes n in the favored direction D with $f_D(n) \leq f\text{Lim}$ and $g_D(n) < g\text{Lim}_D$ are expanded before any node n' in direction \bar{D} with $f_{\bar{D}}(n') \leq f\text{Lim}$ and $g_{\bar{D}}(n') = g\text{Lim}_{\bar{D}} - 1$. We denote this tie-breaking constraint as lb -tie-breaking.

5.1 Defining GBFHS_{lb} given dfMM_{lb}

Given dfMM_{lb} with a fraction p , that without loss of generality prioritize the forward direction in the lb -tie-breaking, we define the following split function denoted by $\text{split}(p)$:

$$g\text{Lim}_F = \lceil p \cdot (f\text{Lim} + 1 - \epsilon) \rceil$$

$$g\text{Lim}_B = \lfloor (1 - p) \cdot (f\text{Lim} + 1 - \epsilon) \rfloor$$

$\text{split}(p)$ is used in the following theorem.

Theorem 4. *Given dfMM_{lb} with a fraction p , GBFHS_{lb} using $\text{split}(p)$ expands nodes n in direction D if and only if $pr_D(n) \leq f\text{Lim}$.*

To prove this theorem the following two lemmas are used.

Lemma 5. *Given dfMM_{lb} with a fraction p , if $pr_D(n) > f\text{Lim}$ for a node n in direction D , then GBFHS_{lb} using $\text{split}(p)$ does not expand n .*

Proof. Let n be a node in the direction D with $pr_D(n) > f\text{Lim}$. Thus, either $f_{D_{lb}}(n) > f\text{Lim}$ or $\frac{g_D(n)}{q_D} + \epsilon > f\text{Lim}$. We will tackle these two cases in their respective order.

Case 1: $f_{D_{lb}}(n) > f\text{Lim}$. Since GBFHS_{lb} only expand nodes with $f_{D_{lb}}(n) \leq f\text{Lim}$, it does not expand n .

Case 2: $\frac{g_D(n)}{q_D} + \epsilon > f\text{Lim}$. Since we have already proven in Case 1 that nodes n with $f_{D_{lb}}(n) > f\text{Lim}$ are not expanded by GBFHS_{lb} , we can further limit Case 2 to nodes for which $f_{D_{lb}}(n) \leq f\text{Lim}$.

Assume in contradiction that GBFHS_{lb} can expand n . Since Case 2 is limited to nodes with $f_{D_{lb}}(n) \leq f\text{Lim}$, GBFHS_{lb} can expand n if and only if $g_D(n) < g\text{Lim}_D$. Therefore, using the definition of $\text{split}(p)$, $g_D(n) < \lceil q_D \cdot (f\text{Lim} + 1 - \epsilon) \rceil$. Using the integer-cost assumption, $g_D(n) \leq \lceil q_D \cdot (f\text{Lim} + 1 - \epsilon) \rceil - 1$. Since $\lceil q_D \cdot (f\text{Lim} + 1 - \epsilon) \rceil \leq q_D \cdot (f\text{Lim} - \epsilon) + 1$, $g_D(n) \leq q_D \cdot (f\text{Lim} - \epsilon) + 1 - 1 = q_D \cdot (f\text{Lim} - \epsilon)$. The previous inequality can be combined with the assumption of Case 2, $\frac{g_D(n)}{q_D} + \epsilon > f\text{Lim}$, to get the following inequality: $\frac{q_D \cdot (f\text{Lim} - \epsilon)}{q_D} + \epsilon > f\text{Lim}$. This results in a contradiction, $f\text{Lim} > f\text{Lim}$. Therefore, GBFHS_{lb} does not expand n . \square

Lemma 6. *Given dfMM_{lb} with a fraction p , if for node n , $pr_D(n) \leq f\text{Lim}$, then GBFHS_{lb} using $\text{split}(p)$ expands n in direction D .*

Proof. Let n be a node in the direction D with $pr_D(n) \leq f\text{Lim}$. Thus $f_{D_{lb}}(n) \leq f\text{Lim}$ and $\frac{g_D(n)}{q_D} + \epsilon \leq f\text{Lim}$.

Case 1: D is forward. $\frac{g_F(n)}{p} + \epsilon \leq f\text{Lim}$, therefore, $g_F(n) \leq p \cdot (f\text{Lim} - \epsilon) < \lceil p \cdot (f\text{Lim} + 1 - \epsilon) \rceil = g\text{Lim}_F$. Thus, n is expanded by GBFHS_{lb} .

Case 2: D is backward. First, observe that if $g_B(n) > g\text{Lim}_B$ (and therefore $g_B(n) \geq g\text{Lim}_B + 1$):

$$pr_B(n) \geq \frac{g\text{Lim}_B + 1}{1 - p} + \epsilon$$

$$= \frac{\lfloor (1 - p) \cdot (f\text{Lim} + 1 - \epsilon) \rfloor + 1}{1 - p} + \epsilon$$

$$> \frac{(1 - p) \cdot (f\text{Lim} - \epsilon)}{1 - p} + \epsilon$$

$$= f\text{Lim}$$

Thus, $g_B(n) \leq g\text{Lim}_B$. If $g_B(n) < g\text{Lim}_B$, n is expanded by GBFHS_{lb} . Otherwise, $g_B(n) = g\text{Lim}_B$, and due to the lb -tie-breaking, there are no nodes n' in Open_F with $f_F(n') \leq f\text{Lim}$ and $g_F(n') < g\text{Lim}_F$. Let n' be the node in Open_F for which $lb(n) = lb(n, n')$. Since $pr_B(n) \leq f\text{Lim}$, and $pr_B(n) \geq lb(n) \geq g_F(n') + g_B(n) + \epsilon$, we know that $g_F(n') + g_B(n) + \epsilon \leq f\text{Lim}$. Moreover, since the minimal g -value for any nodes n' in Open_F with $f_F(n') \leq f\text{Lim}$ is $g\text{Lim}_F$, $g\text{Lim}_F + g_B(n) + \epsilon \leq g_F(n') + g_B(n) + \epsilon$. In addition, $g\text{Lim}_F + g_B(n) + \epsilon - 1 < g\text{Lim}_F + g_B(n) + \epsilon$. Thus, $g\text{Lim}_F + g_B(n) + \epsilon - 1 < f\text{Lim}$. By isolating $g_B(n)$,

we get that $g_B(n) < fLim - \epsilon + 1 - gLim_F$. Therefore, since $fLim = gLim_F + gLim_B + \epsilon - 1$, $g_B(n) < gLim_B$, in contradiction to the result that $g_B(n) = gLim_B$. \square

The proof of Theorem 4 is immediate using the above two lemmas. In addition, since Theorem 4 holds for any value of $fLim$, it follows that GBFHS_{lb} with the proposed split function always expands nodes with minimal priority value, and the minimal priority equals $fLim$ at any given time. Note that any tie-breaking that obeys the *lb*-tie-breaking constraint can be used by both GBFHS_{lb} and dfMM_{lb}. As a result, both GBFHS_{lb} with the suggested split function and dfMM_{lb} expand the same set of nodes in the same order of expansion. Finally, since GBFHS_{lb} and dfMM_{lb} have the same stopping criteria when the minimal priority value is equal to $fLim$, both algorithms terminate at the same time.

5.2 Defining dfMM_{lb} given GBFHS_{lb}

We now proceed to the task of defining an appropriate dynamic fraction p for dfMM_{lb}. Given GBFHS_{lb} with *lb*-tie-breaking that, without loss of generality, prioritizes the forward side, and with a split function $\langle gLim_F, gLim_B \rangle$ for every value of $fLim$, we define a fraction p as follows:

$$p(gLim_F, gLim_B) = \frac{gLim_B}{gLim_F + gLim_B - 1}$$

Thus, for the opposite side,

$$1 - p(gLim_F, gLim_B) = \frac{gLim_F - 1}{gLim_F + gLim_B - 1}$$

Using this fraction, we prove the following Theorem.

Theorem 7. *Given GBFHS_{lb} with a split $\langle gLim_F, gLim_B \rangle$ for every value of $fLim$, dfMM_{lb} with $p(gLim_F, gLim_B)$ expands a node n if and only if $(f_{D_{lb}}(n) \leq fLim$ and $g_D(n) < gLim_D$).*

Proof. Given $fLim$ and a split $\langle gLim_F, gLim_B \rangle$ such that $fLim = gLim_F + gLim_B + \epsilon - 1$, we want to make sure that the priority functions of dfMM_{lb} maintain the behavior of GBFHS_{lb}. Thus, we require all of the following conditions:

1. Nodes n in direction D with $f_{D_{lb}}(n) > fLim$ need to have $pr_D(n) > fLim$;
2. Nodes n in direction D with $g_D(n) \geq gLim_D$ need to have $pr_D(n) > fLim$; and
3. Nodes n in direction D with $g_D(n) < gLim_D$ and $f_{D_{lb}}(n) \leq fLim$ need to have $pr_D(n) \leq fLim$.

We will now show that the all of the above conditions are satisfied.

Condition 1. Since $pr_D(n) \geq f_{D_{lb}}(n)$, this condition holds for any fraction p .

Condition 2. Case 1: D is forward. Since $pr_F(n) \geq \frac{g_F(n)}{p} + \epsilon$,

$$\begin{aligned} pr_F(n) &\geq \frac{g_F(n)}{\frac{gLim_F - 1}{gLim_F + gLim_B - 1}} + \epsilon \\ &= \frac{g_F(n) \cdot (gLim_F + gLim_B - 1)}{gLim_F - 1} + \epsilon \end{aligned}$$

Since $g_F(n) \geq gLim_F$, $\frac{g_F(n) \cdot (gLim_F + gLim_B - 1)}{gLim_F - 1} + \epsilon \geq \frac{gLim_F \cdot (gLim_F + gLim_B - 1)}{gLim_F - 1} + \epsilon$. In addition, since $\frac{gLim_F \cdot (gLim_F + gLim_B - 1)}{gLim_F - 1} + \epsilon > gLim_F + gLim_B - 1 + \epsilon = fLim$, we know that $pr_F(n) > fLim$.

Case 2: D is backward. Assume by contradiction that $pr_B(n) \leq fLim$. Thus, there exists a node n' in $Open_F$ such that $f_{F_{lb}}(n') \leq fLim$ and $g_F(n') + g_B(n) + \epsilon \leq fLim$. Since $g_B(n) \geq gLim_B$, $g_F(n') + g_B(n) + \epsilon \geq g_F(n') + gLim_B + \epsilon$. Thus, $g_F(n') + gLim_B + \epsilon \leq fLim$. Using the definition of $fLim$, $g_F(n') + gLim_B + \epsilon \leq gLim_F + gLim_B + \epsilon - 1$. By isolating $g_F(n')$, we get that $g_F(n') \leq gLim_F - 1$. Therefore,

$$\frac{g_F(n')}{p(gLim_F, gLim_B)} \leq \frac{gLim_F - 1}{\frac{gLim_F - 1}{gLim_F + gLim_B - 1}} = fLim$$

According to the *lb*-tie-breaking, n' should have been expanded before n , in contradiction to $n' \in Open_F$.

Condition 3. Let n be a node in the direction D . Condition 3 dictates that if $g_D(n) < gLim_D$ and $f_{D_{lb}}(n) \leq fLim$, then $pr_D \leq fLim$. Since $f_{D_{lb}}(n) > fLim$, we only need to verify that $\frac{g_D(n)}{q_D} + \epsilon \leq fLim$.

Observe that the expression $\frac{g_D(n)}{q_D} + \epsilon$ is monotonically increasing with the value of $g_D(n)$, therefore, it is sufficient to demand that $\frac{g_D(n)}{q_D} \leq fLim$ for nodes with $g_D(n) = gLim_D - 1$. Thus, we require the following two inequalities to hold:

$$\frac{gLim_F - 1}{p} + \epsilon \leq fLim \quad (1)$$

$$\frac{gLim_B - 1}{1 - p} + \epsilon \leq fLim \quad (2)$$

Substituting $fLim$ with $gLim_F + gLim_B + \epsilon - 1$ in inequality 1 results in the following inequality:

$$p \geq \frac{gLim_F - 1}{gLim_F + gLim_B - 1}$$

Similarly, performing the same substitution in inequality 2 results in the following inequality:

$$p \leq \frac{gLim_F}{gLim_F + gLim_B - 1}$$

Thus, choosing $p = \frac{gLim_F - 1}{gLim_F + gLim_B - 1}$ satisfy both inequalities, and therefore condition 3.

□

Similarly to Theorem 4, Theorem 7 holds for any value of $fLim$. Therefore GBFHS_{lb} with the proposed split function always expands nodes with minimal priority value, and that minimal priority equals $fLim$ at any given time. As a result, dfMM_{lb} with the suggested fraction and GBFHS_{lb} expand the same set of nodes in the same order of expansion. Finally, since GBFHS_{lb} and dfMM_{lb} have the same stopping criteria when the minimal priority value is equal to $fLim$, both algorithms terminate at the same time.

6 Analysis of Algorithm Implementations

We have explored the minimal number of nodes expansion (MNE) for fMM and GBFHS variants. In this section we discuss implementation aspects of the different algorithms.

fMM with a fixed fraction was originally implemented using a priority queue that contains all nodes from both open lists. This implementation can be improved by clustering together nodes that have similar f - and g - values and using the priority queue on the set of clusters instead of the set of nodes. This technique, known as g - f -buckets, was originally suggested for unidirectional heuristic search by Burns et al. (2012) and adapted for Bi-HS by Barley et al. (2018); Shperberg et al. (2019a); and Shperberg et al. (2019b). Using g - f -buckets, each expansion requires an overhead of $O(\log L)$ where L is the number of clusters in G (bounded by the number of states $|V|$), and memory of $O(|V|)$.⁵

Supporting dynamic fractions requires additional overhead. Updating the fraction requires to change the priority of every cluster in the priority queue every time the minimal priority value is changed in a given direction. Since the number of clusters in the open list at any given time is bounded by L , the entire update process is bounded by $O(L)$ time (the time required for updating the cluster priorities and reconstructing the heap). In addition, since the minimal priority value is a lower-bound on the solution, the number of different minimal priorities is *less than or equal to* C^* . Moreover, the number of different minimal priorities is also bounded by the number of clusters, therefore, there could be at most $O(\min\{C^*, L\})$ priority queue updates. Thus, the additional overhead of fMM with dynamic fraction compared to fMM with fixed fraction is bounded by $O(\min\{L \cdot C^*, L^2\})$.

Finally, fMM_{lb} requires additional information of nodes in the opposite direction to compute the lb values; this lb computation costs $O(L)$ time per update. Here too, the lb values need to be updated only $O(\min\{C^*, L\})$ times. Therefore, this addition also induces an overhead of $O(\min\{L \cdot C^*, L^2\})$ compared to fMM.

Barley et al. (2018) did not mention any implementation details when introducing GBFHS. In practice, GBFHS finds

⁵The actual overhead is induced by the number of clusters in OPEN at every given moment, not by the number of clusters in G . However, it is harder to reason about the number of clusters in the open list, as it depends on the search process and on non-trivial characteristics of the problem instance. In addition, after retrieving the cluster with the minimal priority, the entire cluster needs to be expanded before the minimal priority changes. Thus, the amortized cost of all expansions is $O(|V| + L)$ instead of $O(|V| \cdot \log(L))$.

the set of expandable nodes in each iteration. The task of finding the set of expandable nodes is analogous to that of updating the priorities of nodes when updating the dynamic fraction of fMM. Consequently, the computational effort of GBFHS and fMM with dynamic fraction is similar. Moreover, since the task of performing the lb -propagation and the task of finding the set of expandable nodes in GBFHS require a similar computational effort, the complexity of GBFHS_{lb} is comparable to the complexity of GBFHS. As a result, to the best of our knowledge, the implementations of fMM_{lb} and GBFHS_{lb} require similar computational resources, so they are equal in that aspect as well.

7 Summary and Conclusions

We analysed GBFHS and fMM, two parametric bidirectional heuristic search algorithms. We showed that when both algorithms aim to meet at the same point, the MNE value of GBFHS is at least as small as the MNE value of fMM, and for some instances it was shown to be strictly smaller. Therefore GBFHS is said to dominate fMM with respect to MNE. We then showed that dfMM_{lb} dominates GBFHS and fMM, and that dfMM_{lb} and GBFHS_{lb} have the same MNE value for every instance. We then showed a straightforward mapping between dfMM_{lb} and GBFHS_{lb} under which both algorithms expand the same sequence of nodes for any problem instance. Finally, we examined the complexity of existing implementations of all of the above algorithms and deemed that dfMM_{lb} and GBFHS_{lb} require comparable computational resources.

The equivalence between dfMM_{lb} and GBFHS_{lb} indicates that all theoretical properties proven for one algorithm hold for both algorithms. Specifically, as proven for fMM, GBFHS_{lb} with the right split function is now known to be optimally efficient (when considering admissibility on I_{AD} given problem instance from I_{CON} , as mentioned in Section 2.1). In addition, as proven for GBFHS, dfMM_{lb} is guaranteed to halt after finding the first solution in unit edge cost domains. Finally, both GBFHS_{lb} and dfMM_{lb} are reasonable and well-behaved.

Which algorithm should one use? From a pedagogical point of view one might claim that GBFHS_{lb} is simpler to understand and to reason about. The reason is that its iterative-deepening-based structure is relatively intuitive. This is in contrast to the priority function of fMM which has a term that uses $\frac{g(n)}{p}$ (and $\frac{g(n)}{1-p}$) which is not easy to grasp when first encountered. In addition fMM_{lb} needs to update the priorities and this adds pedagogical complexity. On the other hand, the fMM family of algorithms has a known best-first search structure, while the expansion structure of GBFHS and GBFHS_{lb} is less intuitive as it contains the notion of *expandable nodes*. The choice of the algorithm is of course a matter of personal taste/opinion. Nevertheless, it is important that no matter which algorithm one chooses, the similarity between these algorithms is known.

Acknowledgements

This work was supported by Israel Science Foundation (ISF) grant #844/17 to Ariel Felner and Eyal Shimony, by BSF

grant #2017692, by NSF grant #1815660 and by the Frankel center for CS at BGU. Finally, we thank the anonymous reviewers whose comments have greatly improved this paper.

References

- Alcazar, V.; Riddle, P.; and Barley, M. 2020. A unifying view on individual bounds and heuristic inaccuracies in bidirectional search. In *AAAI*.
- Barley, M. W.; Riddle, P. J.; Linares López, C.; Dobson, S.; and Pohl, I. 2018. GBFHS: A generalized breadth-first heuristic search algorithm. In *SoCS*, 28–36.
- Burns, E. A.; Hatem, M.; Leighton, M. J.; and Ruml, W. 2012. Implementing fast heuristic search code. In *SoCS*.
- de Champeaux, D., and Sint, L. 1977. An improved bidirectional heuristic search algorithm. *J. ACM* 24(2):177–191.
- Dechter, R., and Pearl, J. 1985. Generalized best-first search strategies and the optimality of A*. *J. ACM* 32(3):505–536.
- 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.
- 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. *Artif. Intell.* 252:232–266.
- Kaindl, H., and Kainz, G. 1997. Bidirectional heuristic search reconsidered. *J. Artif. Intell. Res.* 7:283–317.
- 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.; Hayoun, A.; Felner, A.; Shimony, S. E.; and Sturtevant, N. R. 2019a. Enriching non-parametric bidirectional search algorithms. In *AAAI*.
- Shperberg, S. S.; Felner, A.; Shimony, S. E.; Sturtevant, N. R.; and Hayoun, A. 2019b. Improving bidirectional heuristic search by bounds propagation. In *SoCS*, 106–114.