Comparing Front-to-Front and Front-to-End Heuristics in Bidirectional Search

Lior Siag[1], **Shahaf S. Shperberg**[1], **Ariel Felner**[1], **Nathan R. Sturtevant**[2,3]

[1] Ben-Gurion University of the Negev
[2] University of Alberta
[3]Alberta Machine Intelligence Institute (Amii)
siagl@post.bgu.ac.il, {shperbsh, felner}@bgu.ac.il, nathanst@ualberta.ca

Abstract

Most recent theoretical and algorithmic work in bidirectional heuristic search (BiHS) used *front-to-end* (F2E) heuristics that estimate the distance to the start and goal states. In this paper, we start exploring front-to-front (F2F) heuristics, which estimate the distance between any pair of states. Devising efficient algorithms that use F2F heuristics is a challenging task. Thus, it is important to first understand the benefits of using F2F heuristics compared to F2E heuristics. To this end, we theoretically and experimentally demonstrate that there is a great potential in using F2F heuristics implying that F2F BiHS is a promising area of future research.

1 Introduction

In *bidirectional heuristic search* (BiHS) the aim is to find a least-cost (shortest) path between two states, $start$ and $goal$, by searching simultaneously from both states until the two search frontiers meet. BiHS algorithms use two types of heuristic functions: **(1)** *front-to-end* (F2E) heuristics, which estimate the distance between any given state to $start$ and to $goal$, and **(2)** *front-to-front* (F2F) heuristics, which estimate the distance between any pair of states. F2F heuristics are harder to build and it may be more computationally expensive to integrate them into BiHS algorithms. By contrast, F2F heuristics are more informative than F2E heuristics and result in fewer node expansions. A key challenge is to develop efficient F2F algorithms that exploits this tradeoff and reduce the overall run time compared to F2E algorithms.

The recent theory of *must-expand pairs* (MEPs) (Eckerle et al. 2017) characterizes the set of forward- and backward-node pairs that must be expanded in order to prove solutions' optimality. The MEP theory establishes a theoretical lower-bound on the total number of nodes that must to be expanded during the search for a given problem instance. A prominent line of research was sparked from that theory (Shaham et al. 2017, 2018; Shperberg et al. 2019b,a, 2021; Sturtevant et al. 2020; Alcázar, Riddle, and Barley 2020; Alcázar 2021) and several algorithms have been developed, such as Near-optimal Bidirectional Search (NBS) (Chen et al. 2017).

The MEP theory was initially defined for both F2E and F2F heuristics, but subsequent research has solely focused on F2E algorithms. To the best of our knowledge, no BiHS algorithm based on the F2F MEP theory has been developed, and the lower bounds for F2F heuristics have not been evaluated. Additionally, the original MEP theory only applies to algorithms which must return an optimal solution for any admissible heuristic. Later, Shaham et al. (2018) extended the MEP theory to cases where algorithms guarantee optimal solutions only for consistent heuristics, and can take advantage of this fact. However, this extension only applies to F2E heuristics and not to F2F heuristics.

There are two main contributions in this paper. First, we fill the gap in the MEP theory and extend it to F2F consistent heuristics. Second, a question arises whether effort should be invested towards developing F2F algorithms. In a set of experiments, we use the theory to compute the lower bounds (on node expansions) for F2F algorithms and compare them to the corresponding bounds for F2E algorithms. Additionally, we also implement a naïve version of NBS that uses F2F heuristics and compare it to the existing F2E NBS. All our results show a substantial reduction in terms of node expansions for the F2F variants over the corresponding F2E variants, implying that F2F BiHS is a promising line of future research if data-structure overheads can be overcome.

2 Definitions and Background

A bidirectional heuristic search (BiHS) problem instance $I = (G, start, goal, h)$ is composed of a graph $G = (V, E)$, a $start$ state, a $goal$ state, and a heuristic function h. Let $d(x, y)$ denote the cost of a shortest path between state x and state y in G. The aim is to find a path from $start$ to $goal$ of cost $d(start, goal)$, denoted as C^*. $\epsilon \geqslant 0$ denotes the cost of the least-cost edge. BiHS algorithms search simultaneously from both $start$ and $goal$ until the two search frontiers meet. Consequently, BiHS algorithms typically maintain two open lists, $Open_F$ for the forward search (F) and $Open_B$ for the backward search (B). Given a direction D, g_D, h_D, f_D denote the g-, h-, and f- values of a node in direction D.

There are two types of heuristics in BiHS (Kaindl and Kainz 1997), illustrated in Figure 1. *front-to-end* (F2E) heuristics employ a forward heuristic $h_F : V \rightarrow \mathbb{R}$ that estimates the distance from any state to the goal state and a backward heuristic $h_B : V \rightarrow \mathbb{R}$ that estimates the distance from the start to any state. Naturally, $f_D(n) = g_D(n) + h_D(n)$. A F2E heuristic h is *bi-admissible* iff

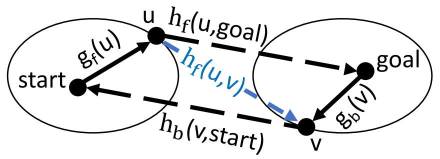

Figure 1: Illustration of F2F and F2E heuristics

$h_F(n) \leqslant d(n, goal)$ and $h_B(n) \leqslant d(start, n)$ for all $n \in V$. In addition, a F2E heuristic h is *bi-consistent* if

(1) $h_F(u) \leqslant d(u, u') + h_F(u')\ for\ all\ u' \in V$
(2) $h_B(v) \leqslant d(v', v) + h_B(v')\ for\ all\ v' \in V$

The second type is *front-to-front* (F2F) heuristics (de Champeaux and Sint 1977), which estimate the distance between *any* pair of states in the graph, $h : V \times V \to \mathbb{R}$. With a F2F heuristic, $h_F(u) = \min_{v \in Open_B}(h(u, v))$ for nodes $u \in Open_F$ and $h_B(v) = \min_{u \in Open_F}(h(u, v))$ for nodes $v \in Open_B$. A F2F is *bi-admissible* iff for all $u, v \in V$, $h(u, v) \leqslant d(u, v)$ and *bi-consistent* iff:

(1) $h(u, v) \leqslant d(u, n) + h(n, v)\ for\ all\ u, v, n \in V$
(2) $h(u, v) \leqslant d(n, v) + h(u, n)\ for\ all\ u, v, n \in V$

Finally, a search algorithm is said to be *Deterministic, Expansion-based, and Black-box* (DXBB) (Eckerle et al. 2017) if it is deterministic, can only discover states, edges, and costs by continuously applying an expansion function from either *start* or *goal*, and has only black-box access to the cost and the heuristic functions.

2.1 Necessary Expansions and GMX

Let I_{AD} denote problem instances that have an admissible heuristic, and let $I_{CON} \subset I_{AD}$ denote the instances that have a consistent heuristic. *Admissible algorithms* are algorithms that guarantee to return optimal solutions on all problem instances from I_{AD}. A classic claim is that any admissible DXBB unidirectional heuristic search algorithm *must expand* all nodes n for which $f(n) < C^*$ to prove the optimality of the solution when running on an instance from I_{CON} (Dechter and Pearl 1985). This classic case is denoted as I_{AD}/I_{CON} where the first part (here I_{AD}) denotes what the algorithm assumes, and the second part (here I_{CON}) is what the algorithm is executed on. In I_{AD}/I_{CON} the algorithm can only assume it is given instances from I_{AD} (consistency is not assumed by the algorithm), and the analysis holds for problem instances from I_{CON}.

This theory of which nodes must be expanded was generalized to BiHS (Eckerle et al. 2017). In BiHS, the *must-expand* attribute is defined for *pairs of nodes* instead of individual nodes. $lb(u, v)$ is a lower-bound on the cost of any path between *start* and *goal* that passes through u and v. For the I_{AD}/I_{CON} case, lb is defined for F2**E** heuristics as:

$$lb^E(u, v) = \max\{f_F(u), f_B(v), g_F(u) + g_B(v) + \epsilon\} \quad (1)$$

and for F2**F** heuristics, lb is defined as:

$$lb^F(u, v) = g_F(u) + g_B(v) + \max(h(u, v), \epsilon) \quad (2)$$

An ordered pair (u, v) is called a *must-expand pair* (denoted MEP) if $lb(u, v) < C^*$ (either lb^E or lb^F). Search algorithms have to expand at least one of the nodes of every MEP, or risk missing the optimal solution. The set of all MEPs can be viewed as a bipartite graph, denoted the *Must-Expand Graph* (GMX) (Chen et al. 2017). For each state $s \in G$, GMX includes a forward node s_F and a backward node s_B. For each pair of states $s, t \in G$, there is an edge between s_F and t_B iff (s, t) is a MEP. Since for all MEPs, at least one node must be expanded, each search algorithm must expand a *vertex cover* of GMX (either GMX$_E$ or GMX$_F$) to prove optimality. Therefore, the minimal node expansions required to guarantee optimality in BiHS is the *minimal vertex cover* (MVC) of GMX. We refer the reader to a survey on MEPs and GMX (Sturtevant and Felner 2018).

2.2 Assuming Consistency

The I_{CON}/I_{CON} case refers to the case where an algorithm can assume that it is only given problem instances with consistent heuristics and is allowed to exploit this knowledge. Shaham et al. (2018) introduced tighter bounds for F2E heuristics for the I_{CON}/I_{CON} case, denoted $lb^{EC}(u, v)$:

$$lb^{EC}(u, v) = g_F(u) + g_B(v) + \max \begin{cases} h_F(u) - h_F(v), \\ h_B(v) - h_B(u), \\ \epsilon \end{cases} \quad (3)$$

Equation 3 can be used for defining a GMX (GMX$_{EC}$).[1] The behavior of F2F algorithms has never been studied for the I_{CON}/I_{CON} case until our current work.

2.3 NBS

Near-Optimal Bidirectional Search (NBS) (Chen et al. 2017) uses F2E heuristics based on the MEP theory. At every expansion, NBS chooses a pair of nodes (u, v) with minimal $lb^E(u, v)$ and expands *both* nodes. NBS in guaranteed to expand at most $2 \times |MVC|$ (of GMX$_E$) to guarantee solution optimality. We denote the original version of NBS as NBS$_E$, since it uses a F2**E** heuristic for the I_{AD}/I_{CON} case. Alcázar (2021) adapted NBS for the I_{CON}/I_{CON} case, denoted here as NBS$_{EC}$. NBS$_{EC}$ often requires fewer expansions before returning an optimal solution, but also induces a significant computational overhead compared to NBS$_E$.

3 Front-to-Front with Consistency

This section develops the notion of MEPs for F2F heuristics for the I_{CON}/I_{CON} case. A straightforward way is to only exploit consistency towards *start* and *goal* (identical to the bi-consistency of F2E heuristics):

(1) $h(u, goal) \leqslant d(u, u') + h(u', goal)\ for\ all\ u' \in V$
(2) $h(start, v) \leqslant d(v', v) + h(start, v')\ for\ all\ v' \in V.$

This results in the following lb (C stands for Consistency):[1]

$$lb^{FC_1}(u, v) = g_F(u) + g_B(v) + \max \begin{cases} h(u, goal) - h(v, goal) \\ h(start, v) - h(start, u) \\ h(u, v) \\ \epsilon \end{cases} \quad (4)$$

[1] If G is known to be undirected, the first two terms in the max term can be replaced with their absolute value.

An even tighter bound on $d(u, v)$ can be defined by considering consistency with respect to all the nodes in the graph, and not only $start$ and $goal$. The bi-consistency definition (Section 2) provides a lower bound on $d(u, v)$ using node $n \in V$. Let $h_1^n(u, v) = h(u, n) - h(v, n) \leqslant d(u, v)$, for some node n, corresponding to the first condition of bi-consistency, and $h_2^n(u, v) = h(n, v) - h(n, u) \leqslant d(u, v)$, corresponding to the second condition of bi-consistency (these are forms of differential heuristics (Sturtevant et al. 2009)). Thus, by taking the node $n \in V$ which maximizes that value, we get the tightest lower-bound on $d(u, v)$ for F2F heuristic for the I_{CON}/I_{CON} case, denoted as lb^{FC_2}:[1]

$$lb^{FC_2}(u, v) = g_F(u) + g_B(v) + \max \begin{cases} \max\limits_{\forall n \in V}\{h_1^n(u, v)\} \\ \max\limits_{\forall n \in V}\{h_2^n(u, v)\} \\ \epsilon \end{cases} \tag{5}$$

The third term from Eq. 4 is redundant because for $n = v$ it holds that $h(u, n) - h(v, n) = h(u, v) - h(v, v) = h(u, v)$. We next prove that lb^{FC_2} can be used for defining MEPs.

Theorem 1. *Given a problem instance $I \in I_{CON}$ with an optimal solution cost of C^*, a forward-optimal path U from start to some state u and a backward-optimal path V from some state v to goal such that $lb^{FC_2}(u, v) < C^*$. Any admissible DXBB algorithm A must expand either u or v when solving instance I. Thus, (u, v) is a MEP.*

Proof. Suppose there is i) a problem instance $I_1 = (G_1 = (V_1, E_1), start, goal, h) \in I_{CON}$ for which the optimal solution has a cost of C_1^*, ii) A forward-optimal path from $start$ to some state u and a backward-optimal path V from some state v to $goal$ such that $lb^{FC_2}(u, v) < C_1^*$, and iii) an admissible DXBB algorithm A that solves I_1 correctly (returns an optimal solution for I_1, $A(I_1)$, with cost C_1^*) without expanding both u and v. In order to prove that A had to expand either u or v (and reach a contradiction), we will construct a problem instance I_2 such that i) $C_{I_2}^* < C_{I_1}^*$, and ii) A also returns $A(I_1)$ when solving I_2 thereby showing that A does not return an optimal solution for I_2 and thus is not admissible, which violates our assumptions and leads to a contradiction. $I_2 = (G_2 = (V_1, E_2), start, goal, h)$, where G_2 is identical to G_1 except for a new edge (u, v), costing $c(u, v) = \max\left(\max_{n \in V_1} h_1^n(u, v), \max_{n \in V_1} h_2^n(u, v), \epsilon\right)$.

This new edge creates a solution path $P = UV$ whose cost is $C_{I_2}^* = lb^{FC_2}(u, v) < C_{I_1}^*$ (assumption iii). A is DXBB and will behave on I_2 exactly as it did on I_1. So, A will neither expand u nor v, so it will not discover edge (u, v), and will incorrectly return $A(I_1)$ of cost $C_{I_1}^* > C_{I_2}^*$.

To conclude the proof, we next show that $I_2 \in I_{CON}$. This is proven by assuming that requirement 1 of bi-consistency is violated and then reaching a contradiction. The proof for a violation of requirement 2 of bi-consistency is analogous. Assume by contradiction that there exist states x and y and n such that $h(x, y) > d_{G_2}(x, n) + h(n, y)$. If the optimal path from x to n in G_2 does not go through the new edge (u, v), then $d_{G_2}(x, n) = d_{G_1}(x, n)$ and since $I_1 \in I_{CON}$, $h(x, y) \leqslant d_{G_1}(x, n) + h(n, y)$ in contradiction to the assumption. Otherwise, the shortest path from x to n goes through (u, v), thus $d_{G_2}(x, n) = d_{G_1}(x, u) + c(u, v) + d_{G_1}(v, n)$. The following inequalities can be derived:

$$\begin{aligned}
&h(x, y) > d_{G_1}(x, u) + c(u, v) + d_{G_1}(v, n) + h(n, y) \\
&\quad \geqslant d_{G_1}(x, u) + h(u, m) - h(v, m) + d_{G_1}(v, n) + h(n, y) \\
&\qquad \forall m \in G_1 \text{ (using the first term in the max of Eq. 5)} \\
&\quad \geqslant d_{G_1}(x, u) + h(u, y) - h(v, y) + d_{G_1}(v, n) + h(n, y) \\
&\quad \text{(for } m = y) \\
&\quad \geqslant h(x, y) + h(v, y) - h(v, y) = h(x, y) \\
&\quad \text{(using the bi-consistency of } G_1)
\end{aligned}$$

thus a contraction is reached and $I_2 \in I_{CON}$. □

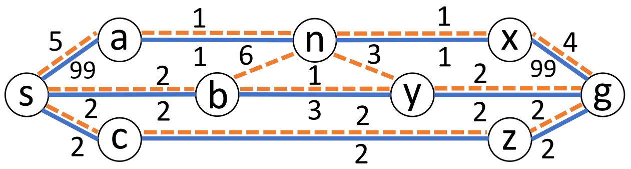

Figure 2: Example of lb^F and lb^{FC_2}

3.1 Example for lb^{FC_2}

Figure 2 shows the advantage of lb^{FC_2} over lb^F. Solid lines are edges and dashed lines are F2F heuristic values between pairs of states. The optimal solution is [s, c, z, g] of cost 6. The key insight is the effect of node n on nodes b and y. Observe that $h(b, y) = 1$ while exploiting consistency via n results in $h_1^n(b, y) = h(b, n) - h(y, n) = 3$.

Assume that s and g were expanded and that $\{a, b, c\}$ are in $Open_F$ and $\{x, y, z\}$ are in $Open_B$. Let $lb_F(u) = \min_{v \in Open_B} lb(u, v)$ and $lb_B(u) = \min_{v \in Open_F} lb(u, v)$ (for any lb definition). Since $h(b, y) = 1$, $lb^F(b, y) = lb_F^F(b) = lb_B^F(y) = 5$.[2] Additionally, $h(c, z) = 2$, thus $lb^F(c, z) = lb_F^F(c) = lb_B^F(z) = 6$. So, algorithms that use lb^F will expand b or y before expanding c or z and finding the optimal path. By contrast, lb^{FC_2} uses $h_1^n(b, y) = 6 - 3 = 3$ and thus $lb^{FC_2}(b, y) = lb_F^{FC_2}(b) = lb_F^{FC_2}(y) = 7$. Thus, algorithms that use lb^{FC_2} can avoid expanding b and y.

Note that lb^{FC_2} is affected by all states in V, so it cannot be fully exploited during search (as the search does not have access to the entire graph). However, we can use a weaker lower bound that only takes the max among states that were discovered during the search. In Figure 2, n, which is the reason for the fact that $lb_F^{FC_2}(b) = lb_B^{FC_2}(y) = 7$, is never discovered during the search. Nevertheless, if we remove nodes a and x then n will be discovered and can be exploited even by the weaker lower bound.

4 Empirical Evaluation

The main purpose of this evaluation is to explore the potential of F2F heuristics compared to F2E heuristics in terms of the number of node expansions, as well as the effect of exploiting consistency on the search.

[2] Subscripts (lb_F) means *forward* while superscripts (lb^F) means F2F. So they can be combined ($lb_F^F(b)$)

		14 Pancake				STP		DAO	
		GAP		GAP-1		MD		Octile	
		$\leqslant C^*$	$< C^*$	$\leqslant C^*$	$< C^*$	$\leqslant C^*$	$< C^*$	$\leqslant C^*$	$< C^*$
F2E	GMX$_E$		39		3,115		531		3,042
	GMX$_{EC}$		37		420		428		2,797
F2F	GMX$_F$		**17**		272		**128**		**2,420**
	GMX$_{FC_1}$		**17**		**268**		**128**		**2,420**

Table 1: Average $|MVC|$ and # of nodes expanded by NBS

	10	15	20	25	30	35	40	45	50
GMX$_E$	8	19	82	236	892	3,888	4,674	11,655	23,012
GMX$_{EC}$	7	19	79	235	853	3,652	4,674	11,655	23,012
GMX$_F$	4	9	27	59	196	482	685	1,305	2,310

Table 2: Average $|MVC|$ with GAP for different #pancakes

	GAP		GAP-1		GAP-2	
	$\leqslant C^*$	$< C^*$	$\leqslant C^*$	$< C^*$	$\leqslant C^*$	$< C^*$
GMX$_{FC_1}$		**3**		**13**		40
NBS$_{FC_1}$	**12**	**4**	28	19	69	60
GMX$_{FC_2}$		**3**		**13**		**31**
NBS$_{FC_2}$	**12**	**4**	**26**	**18**	**55**	**45**

Table 3: Comparing lb^{FC_1} and lb^{FC_2} for 8 pancakes

	STP	DAO	14 Pancake		8 Pancake		
Algorithm	MD	Octile	GAP	GAP-1	GAP	GAP-1	GAP-2
NBS$_E$	0	0	0	0	0	0	0
NBS$_{EC}$	0.1	69.8	0	0.1	0	0	0
NBS$_F$	0.2	67.1	0.9	229.2	0	0	0.1
NBS$_{FC_1}$	0.7	72	4	1,667.9	0	0	0.1

Table 4: Average time for domains in seconds

4.1 Experimental Settings

Domains. We experimented on three domains: **(1) pancake** puzzle with different number of pancakes with the GAP heuristic (Helmert 2010). To get a range of heuristic strengths, we also used the GAP-n heuristics (for $n \in \{1, 2\}$) where the n smallest pancakes are ignored. **(2)** 100 instances of the **8-puzzle** problem (STP) using the Manhattan distance (MD) heuristic. **(3) Grid**-based pathfinding with the octile distance heuristic.We used 156 maps from Dragon Age Origins (DAO) (Sturtevant 2012), each with different start and goal points (a total of 3059 instances). The above are F2F heuristics, and thus can also be used as F2E heuristics.

Metrics. We report the average number of node expansions required to **terminate** (denoted as $\leqslant C^*$). In addition, we report the average number of necessary expansions required to **prove optimality** (denoted as $< C^*$), i.e., the number of nodes expanded until the minimal lb-value (corresponding to the lb-function used by the evaluated algorithm) among all pairs of nodes in Open has reached C^*.

Algorithms. Each lb function (lb^E, lb^{EC}, lb^F, lb^{FC_1}, lb^{FC_2}) induces a corresponding GMX and a variant of NBS, where a pair of nodes from the open list with a minimal lb-value is selected for expansion.

NBS has efficient data structures for the F2E variants of lb (though the I_{CON}/I_{CON} case incurs an additional overhead compared to the I_{AD}/I_{CON} case). The F2F variants are computationally expensive, as the F2F heuristic evaluation in each iteration is quadratic in the number of nodes in Open. Our implementation of NBS$_{FC_2}$ uses only the states that were discovered during the search instead of all the states in the graph (which are unknown to the algorithm).

4.2 Results

Table 1 reports results for the different lb functions, except for lb^{FC_2} which is too expensive to compute. Indeed, the number of necessary expansions of NBS was never more than twice the corresponding size of the MVC. The results show that the theoretical lower-bound (MVC of the GMX), as well as the number of expanded nodes by the corresponding NBS for the F2F heuristics are smaller than those of the F2E heuristics by a factor of up to 2 in the 14-pancake and 8-STP, and 1.4 for DAO (whether or not consistency is assumed). Moving from I_{AD}/I_{CON} to I_{CON}/I_{CON} is very beneficial for the F2E heuristic (up to 7x) but causes almost no improvement for F2F, when considering lb^{FC_1}. In fact, in MD, GAP, and Octile there could be no improvement by utilizing consistency (even with full consistency information as in lb^{FC_2}) since $h(u, v) \geqslant h(u, n) - h(v, n)$ for all $n \in V$. The only exception is for GAP-1 for which the additional consistency information can improve the heuristic.

Table 2 compares the average $|MVC|$ of different GMXs with larger number of pancakes, up to 50. Here we see that as the domain grows, the potential benefit of utilizing F2F heuristics also grows, up to a factor of 10.

To provide results for lb^{FC_2}, we experimented on the 8-pancake problem. The results, reported in Table 3, show that $|MVC|$ of GMX$_{FC_2}$ is smaller than that of GMX$_{FC_1}$ only for GAP-2. This improvement is also evident when comparing NBS$_{FC_1}$ to NBS$_{FC_2}$, even though NBS$_{FC_2}$ utilizes consistency (Eq. 2) based only on the states discovered during the search (and not all states in the graph).

The runtime of the different algorithms is reported in Table 4. Unsurprisingly, the computation overhead of the evaluated F2F variants of NBS is significant, especially when exploiting consistency (though it could be reduced by a more efficient implementation). This indicates that further research is needed to find algorithms that limit the number of F2F heuristic evaluations to balance the reduction in node expansions and the incurred computational overhead.

5 Summary and Conclusions

We have shown that there is potential for using F2F heuristics in BiHS. However, applying F2F heuristics to existing algorithms significantly increase their runtime. Future research should thus determine when to use F2F heuristics for better estimates while controlling the computational overhead. In addition, we are currently studying the idea of exploiting consistency attributes to improve heuristics (i.e., of $h_1^n(u, v)$ and $h_2^n(u, v)$) for unidirectional search algorithms.

Acknowledgements

This research was supported by The Israel US Binational Science Foundation (BSF) grant 2021643 to Ariel Felner.

This work was partially funded 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. AAAI Press.

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. AAAI Press.

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. ijcai.org.

de Champeaux, D.; and Sint, L. 1977. An improved bidirectional heuristic search algorithm. *Journal of the ACM (JACM)*, 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.

Helmert, M. 2010. Landmark Heuristics for the Pancake Problem. In *SoCS*, 109–110. AAAI Press.

Kaindl, H.; and Kainz, G. 1997. Bidirectional heuristic search reconsidered. *Journal of Artificial Intelligence Research*, 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. AAAI Press.

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

Shperberg, S. S.; Felner, A.; Shimony, S. E.; Sturtevant, N. R.; and Hayoun, A. 2019a. Improving Bidirectional Heuristic Search by Bounds Propagation. In *SoCS*, 106–114.

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. AAAI Press.

Sturtevant, N. R. 2012. Benchmarks for Grid-Based Pathfinding. *IEEE Trans. Comput. Intell. AI Games*, 4(2): 144–148.

Sturtevant, N. R.; and Felner, A. 2018. A Brief History and Recent Achievements in Bidirectional Search. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI)*, 8000–8007.

Sturtevant, N. R.; Felner, A.; Barer, M.; Schaeffer, J.; and Burch, N. 2009. Memory-Based Heuristics for Explicit State Spaces. In Boutilier, C., ed., *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 11-17, 2009*, 609–614.

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