Minimizing Node Expansions in Bidirectional Search with Consistent Heuristics

Eshed Shaham
School of Engineering and CS
Hebrew University of Jerusalem
Jerusalem, Israel
eshed.shaham@mail.huji.ac.il

Ariel Felner
ISE Department
Ben-Gurion University
Be'er-Sheva, Israel
felner@bgu.ac.il

Nathan R. Sturtevant
CS Department
University of Denver
United States
sturtevant@cs.du.edu

Jeffrey S. Rosenschein
School of Engineering and CS
Hebrew University of Jerusalem
Jerusalem, Israel
jeff@cs.huji.ac.il

Abstract

A* is optimally effective with regard to node expansions among unidirectional *admissible algorithms*—those that only assume that the heuristic used is admissible. Among bidirectional algorithms the Fractional MM algorithm is optimally effective (given the correct parameters) among *admissible algorithms*. This paper generalizes the bidirectional result to more complex settings where more information on the problem domain can be exploited: (1) When the cost of the minimal edge ϵ is known. (2) When the algorithm knows that the heuristics are consistent. This characterization uses a novel algorithm called `MT`. `MT` is similar to Fractional MM and is also optimally effective, but simpler to analyze.

1 Introduction and Overview

A shortest-path problem, P, is defined as an n-tuple $(G = \{V, E\}, start, goal, h_F, h_B)$. G is a graph, $start, goal \in V$ and the aim is to find the least-cost path (with cost C^*) between $start$ and $goal$. Bidirectional search algorithms interleave two separate searches, a search forward from $start$ and a search backward from $goal$. f_F, g_F and h_F indicate f-, g-, and h-costs in the forward search and similarly f_B, g_B and h_B in the backward search.

Recent research has studied the minimal work (measured by node expansions) required to find and prove an optimal solution, characterizing the conditions for necessary node expansions in *front-to-end* and *front-to-front* search (Eckerle et al. 2017). Subsequent work reformulated these conditions as a *must-expand graph* (G_{MX}), showing that the minimum vertex cover of this graph corresponds to the minimum number of necessary expansions (Chen et al. 2017). Finally, the structure of G_{MX} was analyzed to understand the nature of the minimum vertex cover. The *fractional MM* (`fMM`(p)) algorithm was developed that will expand exactly the minimum vertex cover, given the correct parameter, p, which controls the fraction of the optimal path at which the forward and backward frontiers meet (Shaham et al. 2017).

In this paper we generalize the theory behind `fMM` to other problem settings. First, we present the *meet at the threshold* algorithm (`MT`) which is an algorithm similar to `fMM`. `MT` takes a threshold t, and guarantees that the two bidirectional frontiers will meet at distance t from the start and $C^* - t$ from the goal. Next, we extend the theory to other problem settings and generalize `MT` accordingly. Let I_{AD} be the set of solvable problem instances in which the heuristics are admissible, and let I_{CON} be the subset of problems in I_{AD} where the heuristics are consistent. *Admissible algorithms* are algorithms that must be able to optimally solve *all* instances from I_{AD}. Such algorithms have no further knowledge about the problem and/or the heuristic, and they cannot exploit such knowledge even if it is available. The assumptions made by recent papers on bidirectional search, following the classic analysis of A* (Dechter and Pearl 1985), have been to analyze admissible algorithms running on problem instances with consistent heuristics. That is, the behavior of admissible algorithms was only studied when running on instances from I_{CON}. We denote these assumptions as the *base case* and label it by I_{AD}/I_{CON} along the following notation: *What can be assumed by the algorithm on the problem instances / The instances the algorithm is executed on.*

The assumptions in the base case (I_{AD}/I_{CON}) are limited—the heuristics on the problem instances are consistent but the algorithms cannot exploit that. In this paper we relax these strict assumptions and study the minimum vertex cover of G_{MX} in the following two cases:

Case 1: The algorithm is given the cost of the smallest edge ϵ and is allowed to exploit that knowledge. We label this case as $I_{AD\epsilon}/I_{CON\epsilon}$ and show the adaptations that are needed to generalize I_{AD}/I_{CON} to $I_{AD\epsilon}/I_{CON\epsilon}$.

Case 2: The algorithm is allowed to exploit the fact that both the forward and backward heuristics (h_F and h_B) are consistent. We label this case I_{CON}/I_{CON}. In this case algorithms can assume that they are running on instances from I_{CON} and can exploit a *front-to-front* heuristic that is induced by the consistency of the underlying heuristics. We study the structure of the minimum vertex cover of G_{MX} for this case by partitioning the state space into *equivalence classes* based on h_F and h_B values. We describe a generalization of `MT` for this setting, which requires a unique threshold for each equivalence class. As in `fMM`, the thresholds in `MT` are not known *a priori*. But we prove that there exists a function of special characteristics on the 2-dimensional Euclidean space defined by h_F and h_B, that defines the thresholds of the minimum vertex cover for each *equivalence class*.

Copyright © 2018, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

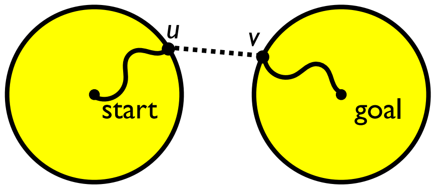

Figure 1: A path from $start$ to $goal$ that goes via u and v

2 Background

Let $d(x, y)$ denote the shortest distance between x and y, so $d(start, goal) = C^*$. The *forward heuristic*, h_F, is *forward admissible* iff $h_F(u) \leq d(u, goal)$ for all u in G and is *forward consistent* iff $h_F(u) \leq d(u, u') + h_F(u')$ for all u and u' in G. The *backward heuristic*, h_B, is *backward admissible* iff $h_B(v) \leq d(start, v)$ for all v in G and is *backward consistent* iff $h_B(v) \leq d(v', v) + h_B(v')$ for all v and v' in G. *Front-to-end* algorithms use these two heuristic functions. Front-to-front bidirectional search algorithms use heuristics between pairs of states on opposite frontiers. See Holte et al. (2017) for a survey of such algorithms.

2.1 Necessary Expansions in Bidirectional Search

With an admissible heuristic, any admissible unidirectional algorithm must expand all states with $f(n) < C^*$ in order to prove the optimal solution (Dechter and Pearl 1985).

Eckerle et al. (2017) generalized this to bidirectional search, showing that necessary expansions are defined on pairs of states u and v in the forward and backward frontiers, respectively. Here we need to reason whether there can be an optimal path that goes from $start$ to u to v to the $goal$, as depicted in Figure 1. Three conditions are required to reason about potential paths between u and v:

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

If all conditions are met, the search must explore to see if there is a shorter path between u and v.

Definition 1. *For each pair of states (u, v) let*

$$lb(u, v) = \max\{f_F(u), f_B(v), g_F(u) + g_B(v)\}$$

In bidirectional search, a pair of states (u, v) is called a *must-expand pair* if $lb(u, v) < C^*$. Unlike unidirectional search, in a *must-expand pair* only one of u or v must be expanded, not both. An algorithm that does not expand either u or v when $lb(u, v) < C^*$ cannot be admissible because it may not find an optimal solution.

2.2 The Must-Expand Graph (G_{MX})

The unique property of the must-expand condition is that it contains an *or*, also a feature of the vertex cover problem. Thus, the conditions for necessary expansions can be represented as the problem of finding a vertex cover on the *must expand graph* denoted by G_{MX} (Chen et al. 2017).

Definition 2. *The Must-Expand Graph G_{MX} on a problem instance is an undirected, unweighed bipartite graph. For each state $u \in G$, there are two vertices in G_{MX}, the left vertex u_F and the right vertex u_B. For each pair of states $u, v \in G$, there is an edge in G_{MX} between u_F and v_B if and only if $lb(u_F, v_B) < C^*$. Thus, there is an edge in G_{MX} between u_F and v_B if and only if the pair (u, v) is a must-expand pair. G_{MX_F} denotes the forward nodes in G_{MX} and G_{MX_B} denotes the backward nodes in G_{MX}.*

It follows that the minimum number of node expansions required to solve a problem using a bidirectional heuristic search algorithm is determined by the size of the minimum vertex cover of G_{MX} (denoted hereafter as MVC). Naturally, the exact set of vertices in G_{MX} depends on the heuristic used in the problem instance. Therefore, MVC is heuristic dependent as well.

Figure 2(a) shows a G_{MX} graph for a sample 20-pancake problem with $C^* = 13$ and the GAP\2 heuristic (Holte et al. 2017). This heuristic counts all gaps (cases where adjacent pancakes are not consecutive numbers) but ignores any gaps between the 2 smallest pancakes, i.e., does not count the gaps involving pancakes 0 or 1. Each node is labeled internally with its g-cost. The left vertices are sorted by increasing g_F-costs, while right vertices are sorted by decreasing g_B-cost. Additionally, nodes with the same g_F or g_B are merged into a single node, and the *weight* of that node, the total number of merged nodes, is written adjacent to the node.[1] In this ordering, each horizontal pair of states u and v have $g_F(u) + g_B(v) = C^*$, which, in Figure 2(a), is 13. Figure 2(b) simplifies G_{MX} by only drawing a full edge from a left node n to a right node if n has no edges to right nodes with larger g_B. Prefixes of the other edges are also shown. Similar edges are shown for right nodes.

NBS (Chen et al. 2017) is a non-parameterized algorithm that efficiently finds a near-optimal vertex cover of G_{MX}. NBS always expands at most twice the number MVC.

2.3 The Minimum Vertex-Cover (MVC) of G_{MX}

The MVC of G_{MX} has the following properties (Shaham et al. 2017). First, MVC is *contiguous* in each direction. That is, if a node n with $g_F(n) = k$ is in MVC, then all nodes with $g_F(n) \leq k$ in G_{MX} must also be in MVC and similarly for g_B. Second, there exist thresholds t^*_F, t^*_B such that (1) $t^*_F + t^*_B = C^*$ and (2) a forward node u from G_{MX} is included in MVC iff $g_F(u) < t^*_F$, and a backwards node v from G_{MX} is included in MVC iff $g_B(v) < t^*_B = C^* - t^*_F$. In Section 4 we will generalize the proof to Case 1 ($I_{AD\epsilon}/I_{CON\epsilon}$).

To demonstrate this, consider all pairs of values of t_F and t_B where $t_F + t_B = C^*$.[2] There is a family of contiguous vertex covers for all such pairs (t_F, t_B), where all nodes with $g_F < t_F$ in G_{MX} are included in the forward direction of this vertex cover, and all nodes with $g_B < t_B$ are included in the backward direction of this vertex cover. One such (t_F, t_B)

[1] Per the definition of G_{MX}, only nodes with $f_F < C^*$ or with $f_B < C^*$ may be included in a must expand pair. Thus, only such nodes are counted here.

[2] To exclude meaningless pairs, such as those with negative numbers, we require one of t_F or t_B to be equal to the g-value of some node in G_{MX}.

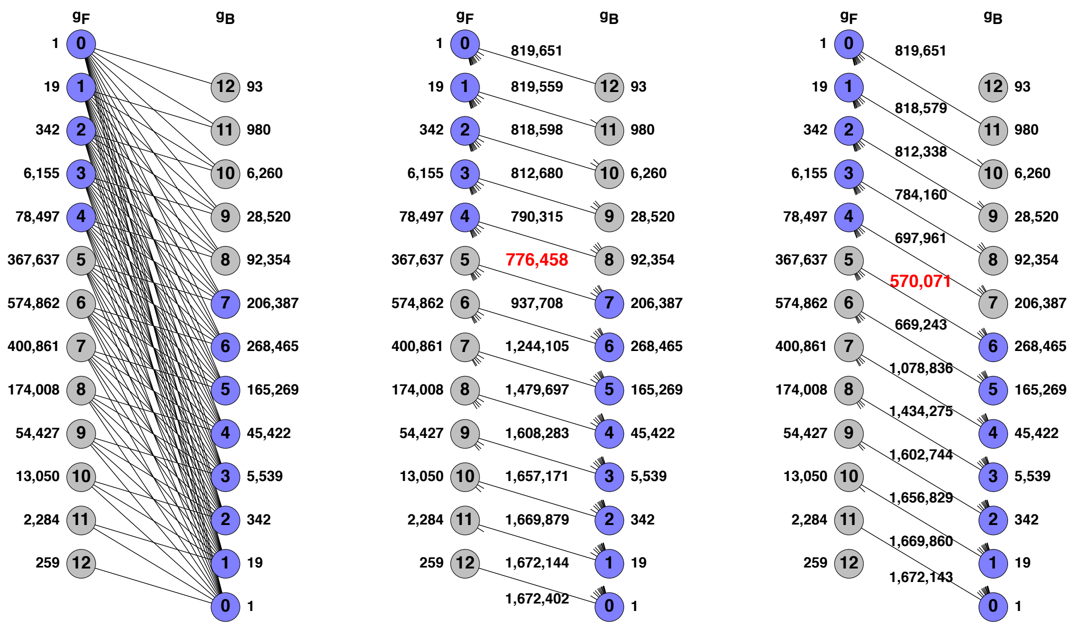

Figure 2: (a) Left: Full G_{MX} (b) Middle: Simplified G_{MX} (c) Right: Simplified G_{MX} for $I_{AD\epsilon}/I_{CON\epsilon}$

partition is the MVC and is denoted by (t_F^*, t_B^*). The cost (in node expansions) of each (t_F, t_B) vertex-cover partitions is determined by summing up the weights of the nodes with $g_F < t_F$ and with $g_B < t_B$. In Figure 2, nodes that correspond to each (t_F, t_B) pair are aligned horizontally. The cost of the the relevant partition is shown between the respective values of t_F and t_B. For example, in Figure 2(b) the $(3, 10)$ partition requires 812,680 nodes. The MVC partition for this example with $(t_F^*, t_B^*) = (5, 8)$ only requires 776,458 node expansions. Nodes that are included in MVC are colored blue, i.e., nodes with $g_F \leq 4$ and with $g_B \leq 7$.

2.4 Fractional MM

MM is a bidirectional search algorithm that is guaranteed to *meet in the middle* (Holte et al. 2017). That is, MM will never expand a state whose g-value exceeds $C^*/2$. *Fractional MM* (fMM) is a generalization of MM that can meet at any fraction of the optimal solution cost (Shaham et al. 2017). fMM(p) uses the following priority functions on paths in the open lists, where $0 < p < 1$:

$$pr_F(u) = \max(g_F(u) + h_F(u), g_F(u)/p)$$
$$pr_B(v) = \max(g_B(v) + h_B(v), g_B(v)/(1-p))$$

fMM(p) expands a state with minimum priority in either direction. fMM(p)'s forward and backward searches meet at $(p \cdot C^*, (1-p) \cdot C^*)$ by similar reasoning for why MM meets in the middle. That is, fMM(p) will never forward expand a state whose g_F-value exceeds $p \cdot C^*$ and never backward expand a state whose g_B-value exceeds $(1-p) \cdot C^*$. This attribute is called *restrained with respect to p*. As a result, a restrained algorithm will always return the optimal solution because it can only meet at the meeting point along the optimal path, along the same reasoning proved for MM (Holte et al. 2017). MM is a special case of fMM(p) for $p = 1/2$. Forward A* and reverse A* (searching from *goal* to *start*) are also special cases corresponding to $p = 1$ and $p = 0$ respectively (although division by zero must be avoided).

For every problem instance, there exists a fraction $0 \leq p^* \leq 1$ such that fMM(p^*) is *optimally effective* and will expand the minimal number of necessary nodes—those in MVC. In practice, however, the exact value for p^* is not known a priori, as it depends on C^* and G_{MX}. But, given C^* and G_{MX} it is possible to find p^* in time linear in the number of distinct g-values. Finding p^* and then running fMM(p^*) can serve for research purposes as an oracle—any algorithm can now be compared to the theoretical optimum.

3 Meet at the Threshold

In this section we describe the *meet at the threshold* algorithm (MT). MT is similar to fMM in the sense that it can meet at any meeting point, but the meeting point is described as a constant threshold and not as a fraction of C^*.

Given a threshold t, MT is *restrained* with respect to t, or equivalently, MT meets at $(t, C^* - t)$. That is, MT will never forward expand a state whose g_F-value exceeds t and never backward expand a state whose g_B-value exceeds $C^* - t$. MT(t) uses the following priority functions:

$$pr_F(u) = \begin{cases} g_F(u) + h_F(u) & \text{if } g_F(u) < t \\ \infty & \text{if } g_F(u) \geq t \end{cases}$$
$$pr_B(v) = \max(g_B(v) + h_B(v), g_B(v) + t)$$

We now show that MT meets at $(t, C^* - t)$. We do this along the same line of proof given for MM (Holte et al. 2017) which uses the fact that there is always a node in $Open_F$ or in $Open_B$ on the optimal path with priority $\leq C^*$.

Lemma 1. *Let u be a forward node. If $g_F(u) > t$ then u will never be forward expanded.*

Proof: By definition, $pr_F(u) = \infty$. □

Lemma 2. *Let v be a backward node. If $g_B(v) > C^* - t$ then $pr_B(v) > C^*$.*

Proof: $pr_B(v) \geq g_B(v) + t > C^* - t + t = C^*$. □

For $t = 0$, $\texttt{MT}(t)$ is identical to backward A*. Similarly, for $t \geq C^*$, $\texttt{MT}(t)$ is identical to forward A*. It is easy to see that $\texttt{MT}(t)$ will expand the same set of nodes in G_{MX} as $\texttt{fMM}(p)$ when $t = p \cdot C^*$ (not necessarily in the same order). For example, MM is identical to $\texttt{MT}(t)$ for $t = C^*/2$. Therefore, for any value of t, MT will return the optimal solution but with the correct value for t ($t^* = p^* \cdot C^*$), $\texttt{MT}(t)$ is *optimally effective* as well.

Similarly to fMM, the best value for t is not known in advance. The usage of p (a relative threshold) to describe fMM and to define vertex covers of G_{MX} rather than t (a constant threshold) resulted from the attempt to generalize existing algorithms, such as MM and A*. However, our proofs below (and in fact, those of fMM too) use a constant threshold. Therefore, it is more natural to use MT in this context.

In the next sections we study G_{MX} and its MVC, and generalize MT (and fMM) to the cases where the algorithm can assume more knowledge in the graph and on h.

4 Case 1: Knowing ϵ ($I_{AD\epsilon}/I_{CON\epsilon}$)

In many cases, the problem instance is coupled with a lower bound (ϵ) on the edge costs. For example, in unit edge-cost domains $\epsilon = 1$. In other domains, one might iterate over all actions costs and take their minimum. This bound provides an admissible front-to-front heuristic of ϵ between any two distinct nodes. We now study the case where algorithms are allowed to assume the existence of ϵ but as done in previous work still can only assume an admissible heuristic. Similarly, we assume that they are evaluated in instances where the heuristic is consistent. In our notation, this case is labeled by $I_{AD\epsilon}/I_{CON\epsilon}$. For this case we adapt the conditions for *must-expand pairs* in domains with front-to-front heuristics as defined by Eckerle et al. (2017) with ϵ as the front-to-front heuristic. In such cases two nodes u, v are a *must-expand pair* if the following conditions are met:

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

The third condition is more informative than the front-to-end version of the base case due to the addition of ϵ (in Figure 1, ϵ is a lower bound on the cost of the path from u to v). Therefore, some edges that exist in the old G_{MX} no longer exist in the new G_{MX}. The simplified G_{MX} of the same pancake instance for the $I_{AD\epsilon}/I_{CON\epsilon}$ case is shown in Figure 2(c). In the pancake puzzle, $\epsilon = 1$ and thus for each node there is one less edge. For example, the forward node with $g_F = 4$ was connected to all backward nodes with $g_B \leq 8$ for the base case (Figure 2(b)) but it is now only connected to nodes with $g_B \leq 7$.

The former characteristics of MVC for the base case (I_{AD}/I_{CON}) were effectively proven with $\epsilon = 0$. We will next prove a few characteristics of MVC and provide an optimally effective algorithm for the $I_{AD\epsilon}/I_{CON\epsilon}$ case. The proofs here are simpler and valid for any non-negative value of ϵ (including $\epsilon = 0$ in the base case). Part of this simplification is the usage of MT. The following lemmas and proofs are for the forward direction; the lemmas and proofs for the backward direction are analogous.

Lemma 3. *Let u be a forward node such that $u \in$ MVC. There exists a backward node v such that (u, v) is a must-expand pair and $v \notin$ MVC.*

Proof: Assume to the contrary that all nodes v' such that (u, v') is a *must-expand pair*, are in MVC. We could then remove u from MVC as all its neighbors are in MVC. This contradicts the minimality of MVC. □

Lemma 4. *Let $u, u' \in G_{MX}$ be two forward nodes such that $u \in$ MVC and $u' \notin$ MVC. Then, $g_F(u') > g_F(u)$.*

Proof: From Lemma 3 we know that there exists a backward node v such that (u, v) is a *must-expand pair* but $v \notin$ MVC. The pair (u', v) cannot be a *must-expand pair* because both u' and v are not in MVC. Therefore, it must hold that $g_F(u') + g_B(v) + \epsilon \geq C^* > g_F(u) + g_B(v) + \epsilon$, yielding $g_F(u') > g_F(u)$. □

Corollary 5. *There exists a threshold t^*_F for which* MVC *contains exactly all forward nodes in G_{MX} where $g_F < t^*_F$.*

This is a direct result of Lemma 4. If all forward nodes in MVC have g_F-values smaller than those not in MVC such a threshold t^*_F must exist.

With similar proofs for the backward directions we now know that MVC can be defined using a pair of thresholds (t^*_F, t^*_B). Next, we prove that $t^*_F + t^*_B + \epsilon = C^*$ and show that such thresholds exist by constructing such a pair.

If MVC only contains forward nodes then $t^*_F = C^* - \epsilon, t^*_B = 0$. Similarly if it only contains backward nodes, then $t^*_F = 0, t^*_B = C^* - \epsilon$. Below we will assume the non-trivial case where nodes from both directions are inside MVC.

Definition 3.

$$g_{F^{\text{I}}} = \max_{u \in \left(G_{\text{MX}_F} \cap \text{MVC}\right)} \{g_F(u)\}$$
$$g_{F^{\text{O}}} = \min_{u \in \left(G_{\text{MX}_F} \setminus \text{MVC}\right)} \{g_F(u)\}$$
$$g_{B^{\text{I}}} = \max_{v \in \left(G_{\text{MX}_B} \cap \text{MVC}\right)} \{g_B(v)\}$$
$$g_{B^{\text{O}}} = \min_{v \in \left(G_{\text{MX}_B} \setminus \text{MVC}\right)} \{g_B(v)\}$$

$g_{F^{\text{I}}}$ is the g_F value of the last forward node inside MVC, $g_{F^{\text{O}}}$ is the g_F value of the first forward node outside MVC. Similarly, $g_{B^{\text{I}}}$ is the last backward node inside MVC, $g_{B^{\text{O}}}$ is the first backward node outside MVC. Now we define the thresholds:

$$t^*_F = \frac{\max\{g_{F^{\text{I}}}, C^* - \epsilon - g_{B^{\text{O}}}\} + \min\{g_{F^{\text{O}}}, C^* - \epsilon - g_{B^{\text{I}}}\}}{2}$$

$$t^*_B = \frac{\max\{g_{B^{\text{I}}}, C^* - \epsilon - g_{F^{\text{O}}}\} + \min\{g_{B^{\text{O}}}, C^* - \epsilon - g_{F^{\text{I}}}\}}{2}$$

With simple arithmetic it is easy to see that $t^*_F + t^*_B + \epsilon = C^*$. Since t^*_F and t^*_B are entirely symmetric in their definition we prove the following Lemmas for the forward direction only. The proofs for the backward directions are symmetric. The inequalities in the following lemma treat each of the four ways to pair a term from the left (max) and right (min) sides of the definition of t^*_F.

Lemma 6. *(helping inequalities)*

1. $g_{F^{\text{I}}} < t^*_F \leq g_{F^{\text{O}}}$
2. $g_{F^{\text{I}}} < t^*_F < C^* - \epsilon - g_{B^{\text{I}}}$
3. $C^* - \epsilon - g_{B^{\text{O}}} \leq t^*_F \leq g_{F^{\text{O}}}$
4. $C^* - \epsilon - g_{B^{\text{O}}} \leq t^*_F < C^* - \epsilon - g_{B^{\text{I}}}$

Proof: Since t^*_F is defined as the mean of the maximum of the left terms and the minimum of the right terms, it is sufficient to prove each inequality only between the left and the right terms.

1. $g_{F^{\text{I}}} < g_{F^{\text{O}}}$
Proof: Follows directly from Lemma 4. □

2. $g_{F^{\text{I}}} < C^* - \epsilon - g_{B^{\text{I}}}$
Proof: From Definition 3 there must exist a forward node $u \in$ MVC and a backward node $v \in$ MVC such that $g_F(u) = g_{F^{\text{I}}}$ and $g_B(v) = g_{B^{\text{I}}}$. According to Lemma 3 we know that there exists a backward node $v' \notin$ MVC such that (u, v') is a *must-expand pair* and $g_F(u) + g_B(v') + \epsilon < C^*$. From Lemma 4 we know that $g_B(v) < g_B(v')$. Therefore, $g_{F^{\text{I}}} + g_{B^{\text{I}}} + \epsilon = g_F(u) + g_B(v) + \epsilon < g_F(u) + g_B(v') + \epsilon < C^*$. □

3. $C^* - \epsilon - g_{B^{\text{O}}} \leq g_{F^{\text{O}}}$.
Proof: We need to show that $g_{F^{\text{O}}} + g_{B^{\text{O}}} + \epsilon \geq C^*$. From Definition 3 there must exist a forward node $u \notin$ MVC and a backward node $v \notin$ MVC such that $g_F(u) = g_{F^{\text{O}}}$ and $g_B(v) = g_{B^{\text{O}}}$. Since both nodes are not in MVC we know that (u, v) is not a *must-expand pair*, therefore $g_{F^{\text{O}}} + g_{B^{\text{O}}} + \epsilon \geq C^*$. □

4. $C^* - \epsilon - g_{B^{\text{O}}} < C^* - \epsilon - g_{B^{\text{I}}}$.
Proof: This is direct result of the fact that $g_{B^{\text{I}}} < g_{B^{\text{O}}}$ (Lemma 4). □

Lemma 7. *For a forward node u, $u \in$ MVC iff $g_F(u) < t^*_F$.*

Proof: If $u \in$ MVC, then directly from Definition 3 we know that $g_F(u) \leq g_{F^{\text{I}}}$ and from Lemma 6 $g_F(u) < t^*_F$. If $u \notin$ MVC then directly from Definition 3 we know that $g_F(u) \geq g_{F^{\text{O}}}$ and from Lemma 6 $g_F(u) \geq t^*_F$. □

This completes the proof that the two thresholds t^*_F and t^*_B, as defined above, describe MVC. For Figure 2(c) we have $t^*_F = 5$ and $t^*_B = 7$ (as opposed to the base case which had $t^*_B = 8$). The nodes included in MVC for this case are colored blue and the number of nodes in that partition is highlighted. It is now only 570,071 compared to 776,458 for the base case. This is because the node with $g_F = 7$ which had a count of 206,387 is no longer included in MVC.

4.1 Meet at the Threshold ϵ (MTϵ)

Lemma 7 implies a simple algorithm for finding MVC given that G_{MX} and C^* are both known. We iterate over all possible thresholds that satisfy $t_F + t_B + \epsilon = C^*$ and sum up the costs of the forward nodes $u \in G_{\text{MX}}$ for which $g_F(u) < t_F$ as well as the costs of the backward nodes $v \in G_{\text{MX}}$ for which $g_B(v) < t_B$. This algorithm runs with complexity linear in the number of distinct g-values of states in G_{MX}. It is similar to the *Calc-VC()* algorithm provided by Shaham et al. (2017) which iterated on all thresholds that hold $t_F + t_B = C^*$. The priority function of MT can easily be adapted to create a new algorithm MTϵ:

$$pr_F(u) = \begin{cases} g_F(u) + h_F(u) & \text{if } g_F(u) < t_F \\ \infty & \text{if } g_F(u) \geq t_F \end{cases}$$
$$pr_B(u) = \max(g_B(u) + h_B(u), g_B(u) + t_F + \epsilon)$$

Similar to MT(t) in the base case, for any given problem instance I in $I_{CON\epsilon}$ there exists t^*_F such that MT$\epsilon(t)$ is optimally effective—it expands the minimal number of nodes in G_{MX} of I. The proof for this is nearly identical to the respective proof on fMM so it is not repeated here.[3]

5 Case 2: Consistency (I_{CON}/I_{CON})

A heuristic on an undirected graph[4] is *forward consistent* if for every two nodes u, v $|h_F(u) - h_F(v)| \leq d(u, v)$. *Backward consistency* is defined analogously.

As explained above, the analysis of bidirectional search by (Eckerle et al. 2017) and subsequent papers (Chen et al. 2017; Shaham et al. 2017) assumed admissible algorithms, i.e., algorithms only know that the heuristic is admissible (I_{AD}). However, they all analyzed the case where such algorithms are executed on instances from I_{CON} (but the algorithms do not know that and cannot exploit that fact). Assuming that execution is on instances from I_{CON} simplifies the analysis, by guaranteeing that no nodes will be re-expanded. However, such a setting is limited. We next investigate the case where the algorithms know that the heuristics are consistent (i.e., they know that the instances are from I_{CON}) and they are allowed to exploit that knowledge. We label this case I_{CON}/I_{CON}.

Given consistent heuristics, the difference $|h_F(u) - h_F(v)|$ is, in fact, a lower bound on the distance between u and v, so $|h_F(u) - h_F(v)|$ can be used as a front-to-front heuristic between u and v. Similarly, $|h_B(u) - h_B(v)|$ can be used as a heuristic as well. This was called the *Add* method by Kaindl and Kainz (1997).

Definition 4. (Front-to-front heuristic) *For each pair of nodes (a, b) define the following admissible heuristic:*

$$h_C(a, b) = \max\{|h_F(a) - h_F(b)|, |h_B(a) - h_B(b)|\}$$

This heuristic results from the combination of assuming both forward and backward consistency (hence h_C).

We next prove a form of the Triangle Inequality for h_C, as it will be used below in many circumstances.

Lemma 8. (Triangle Inequality of h_C) *For any three states a, b, c, $h_C(a, c) + h_C(c, b) \geq h_C(a, b)$*

Proof: For the first part of the proof we show
$h_C(a, c) + h_C(c, b)$
$= \max\{|h_F(a) - h_F(c)|, |h_B(a) - h_B(c)|\}$
$+ \max\{|h_F(c) - h_F(b)|, |h_B(c) - h_B(b)|\}$

[3]The priority function of fMM could be adapted to create a new algorithm fMMϵ in a similar manner to MMϵ (Holte et al. 2017). For fMMϵ: $pr_F(u) = \max(f_F(u), g_F(u)/p + \epsilon)$ and $pr_B(v) = \max(f_B(v), g_B(v)/(1-p) + \epsilon)$.

[4]For simplicity we assume that the graph is undirected; the theory below can be modified to also work for directed graphs.

$\geq |h_F(a) - h_F(c)| + |h_F(c) - h_F(b)|$
$\geq |h_F(a) - h_F(b)|$ *(from the regular triangle inequality)*.
In a similar way we can show that $h_C(a,c) + h_C(c,b) \geq |h_B(a) - h_B(b)|$; thus we can conclude that $h_C(a,c) + h_C(c,b) \geq \max\{|h_F(a) - h_F(b)|, |h_B(a) - h_B(b)|\} = h_C(a,b)$. □

5.1 Necessary Node Expansions

Using the conditions for *must-expand pairs* (Eckerle et al. 2017) in problems with h_C as a front-to-front heuristic results in the following claim: two nodes u, v are a *must-expand pair* if the following conditions are met:

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

Any pair of nodes u, v that satisfy these conditions is a *must-expand pair* and therefore there is an edge in G_{MX} between u and v. The third condition is more informative than the front-to-end version due to the addition of h_C. Therefore, some edges that exist in the old G_{MX} no longer exist in the new G_{MX}. We analyze G_{MX} in this case based on the notion of equivalence classes defined next.

5.2 Equivalence Classes

In the I_{AD}/I_{CON} case $h_C(u,v)$ did not exist. To imitate this in the I_{CON}/I_{CON} case, we partition G_{MX} into equivalence classes, such that the nodes in each class have a front-to-front heuristic of 0 to each other.

Definition 5. (heuristic equivalence) *Two nodes u, v are heuristically equivalent (or h-equivalent) iff $h_C(u,v) = 0$.*

Directly from the definition, two nodes u, v are h-equivalent (belong to the same h-equivalence class) iff $h_F(u) = h_F(v)$ and $h_B(u) = h_B(v)$. In other words, each equivalence class is defined by the pair of heuristic values (h_F, h_B) of the nodes within.

The following lemmas will prove that the intersection of MVC with each equivalence class has similar properties to the base case which effectively only had a single class. That is, MVC is *contiguous and restrained* with respect to t_F and t_B. The proofs are similar to the former two cases, with changes necessary to handle nodes in one equivalence class that have neighbors in other classes, which could not happen when there was only a single class.

Lemma 9. *Let $u, u' \in G_{MX}$ be two h-equivalent forward nodes. If $u \in$ MVC and $u' \notin$ MVC. Then, $g_F(u') > g_F(u)$.*

Proof: From Lemma 3 we know there exists a backward node v such that (u,v) is a must-expand pair but $v \notin$ MVC. The pair (u',v) cannot be a must-expand pair because both u' and v are not in MVC. Therefore, it must hold that $g_F(u') + g_B(v) + h_C(u',v) \geq C^*$. Since u and u' are h-equivalent, $h_C(u,v) = h_C(u',v)$ yielding that $g_F(u') + g_B(v) + h_C(u,v) \geq C^* > g_F(u) + g_B(v) + h_C(u,v)$. As a result, $g_F(u') > g_F(u)$. □

Corollary 10. *For every h-equivalence class Q there exists a threshold t^*_{FQ} such that* MVC *contains exactly all forward nodes in Q where $g_F < t^*_{FQ}$.*

This is a direct result of Lemma 9. If all forward nodes in MVC have g_F-values lower than those not in MVC the described threshold t^*_{FQ} must exist. The proof for the backward direction is identical.

Intermediate Summary: So far we have shown that in each equivalence class Q, there exists a pair of thresholds (t^*_{FQ}, t^*_{BQ}) that impose which nodes from Q are contained in MVC. To show the *restrained* property, we will have to prove that for every h-equivalence class Q there exists a pair of such thresholds (t^*_{FQ}, t^*_{BQ}), that satisfy $t^*_{FQ} + t^*_{BQ} = C^*$ (as $h_C(u,v) = 0$ for every forward node u and backward node v in this class). We will prove this by showing that for any two forward and backward nodes $u, v \in Q$:

1. If $u, v \notin$ MVC then $g_F(u) + g_B(v) \geq C^*$. (Lemma 11)
2. if $u, v \in$ MVC then $g_F(u) + g_B(v) < C^*$. (Lemma 12)

Lemma 11. *Let $u, v \in G_{MX}$ be a forward node and a backward node respectively such that u and v are h-equivalent. If $u \notin$ MVC and $v \notin$ MVC then $g_F(u) + g_B(v) \geq C^*$.*

Proof: Neither of the nodes u, v are in MVC therefore the must-expand condition is false, so it must hold that $g_F(u) + g_B(v) + h_C(u,v) = g_F(u) + g_B(v) \geq C^*$. □

Lemma 12. *Let $u, v \in G_{MX}$ be a forward node and a backward node respectively such that u and v are h-equivalent. If $u \in$ MVC and $v \in$ MVC then $g_F(u) + g_B(v) < C^*$.*

Proof: According to Lemma 3 there exists a backward node $v' \in G_{MX}$ such that (u,v') is a must-expand pair and $v' \notin$ MVC. Similarly there exists a forward node $u' \in G_{MX}$ such that (u',v) is a must-expand pair and $v' \notin$ MVC. We know that $g_F(u) + g_B(v') + h_C(u,v') < C^*$ and $g_F(u') + g_B(v) + h_C(u',v) < C^*$. We also know that $g_F(u') + g_B(v') + h_C(u',v') \geq C^*$ since $u', v' \notin$ MVC. Recall that u and v have identical h-values so together with the triangle inequality $h_C(u,v') + h_C(u',v) \geq h_C(u',v')$. Now $g_F(u) + g_B(v) < C^* - g_B(v') - h_C(u,v') + g_B(v) < 2C^* - g_B(v') - h_C(u,v') - g_F(u') - h_C(u',v) \leq 2C^* - g_B(v') - g_F(u') - h_C(u',v') \leq C^*$. □

Lemma 13. (Restrained with respect to t^*_{FQ}) *For every h-equivalence class Q there exist thresholds $t^*_{FQ} + t^*_{BQ} = C^*$ such that* MVC $\cap\, Q$ *contains exactly the forward nodes in Q for which $g_F < t^*_{FQ}$. and exactly all backward nodes for which $g_B < t^*_{BQ}$.*

Proof: This is the result of Corollary 10, Lemma 11 and Lemma 12. □

5.3 From Equivalence Classes to Functions

The existence of a threshold for every equivalence class implies that there exists a function that maps equivalence classes into thresholds. Recall that each class is uniquely defined by its h_F- and h_B-values, so this function can also be expanded to include every pair of values (h_F, h_B) to two thresholds as long as any pair (h_F, h_B) which has members in G_{MX} will get the respective thresholds. For a pair of values with no members in G_{MX} any threshold will be valid. Therefore, the function can be expanded to all $\mathbb{R}^+ \times \mathbb{R}^+$.

Case	I_{AD}/I_{CON}	$I_{AD\epsilon}/I_{CON\epsilon}$	I_{CON}/I_{CON}
MVC	Restrained	ϵ-Restrained	$b \in B$
Optimal Alg.	`fMM` / `MT`	`MT`ϵ / `fMM`ϵ	$\texttt{MT}_{CON}$
# Clusters	$\lvert\mathbb{G}\rvert$	$\lvert\mathbb{G}\rvert$	$\lvert\mathbb{G}\rvert \times \lvert\mathbb{H}\rvert^2$
Complexity	$O(\lvert\mathbb{G}\rvert)$	$O(\lvert\mathbb{G}\rvert)$	$O(\lvert\mathbb{G}\rvert^2 \times \lvert\mathbb{H}\rvert^4)$

Table 1: A summarizing table. $|\mathbb{G}|$ and $|\mathbb{H}|$ are the numbers of distinct g- and h-values respectively.

Corollary 14. *There exists a function $b : \mathbb{R}^+ \times \mathbb{R}^+ \rightarrow \mathbb{R}^+$ such that a forward node u is in* MVC *iff $g_F(u) \leq b(h_F(u), h_B(u))$ and a backward node v is in* MVC *iff $g_B(u) \leq C^* - b(h_F(u), h_B(u))$.*

Below we use $b(u)$ as shorthand for $b(h_F(u), h_B(u))$. Given the function b as just defined, we can now adapt `MT` to work in the I_{CON}/I_{CON} case. We define the algorithm $\texttt{MT}_{CON}(b)$ with the following priorities:

$$pr_F(u) = \begin{cases} g_F(u) + h_F(u) & \text{if } g_F(u) < b(u) \\ \infty & \text{if } g_F(u) \geq b(u) \end{cases}$$
$$pr_B(u) = \max(g_B(u) + h_B(u), g_B(u) + b(u))$$

5.4 Attributes of b

As discussed above, given any threshold $t_F \in \mathbb{R}^+$, `MT` and `MT`ϵ (and similarly, `fMM` and `fMM`ϵ) will always return an optimal path, although they might do more work than the minimal number of necessary expansions (as achieved by t^*_F). $\texttt{MT}_{CON}$, however, is not guaranteed to return an optimal path for any $b : \mathbb{R}^+ \times \mathbb{R}^+ \rightarrow \mathbb{R}^+$.

Fortunately this issue can be solved by constraining b to be a member of the following set of functions B:

$$B = \{b : \mathbb{R}^+ \times \mathbb{R}^+ \rightarrow \mathbb{R}^+ \mid \forall {h_F}', {h_B}', {h_F}'', {h_B}'' \in \mathbb{R}^+$$
$$\left|b({h_F}', {h_B}') - b({h_F}'', {h_B}'')\right|$$
$$\leq \max\{\left|{h_F}' - {h_F}''\right|, \left|{h_B}' - {h_B}''\right|\}$$

This is a stronger version of the global *1-Lipschitz* attribute on the continuity of functions. In the appendix we prove two theorems:

1. Given a function $b \in B$, $\texttt{MT}_{CON}(b)$ will return an optimal path on any problem instance in I_{CON}. (Theorem 17)
2. Given a problem instance in I_{CON} there exists $b^* \in B$ such that $\texttt{MT}_{CON}(b)$ will expand the minimal number of necessary expansions on that instance. (Theorem 22)

From these theorems we can deduce that $\texttt{MT}_{CON}$ is optimally effective in the I_{CON}/I_{CON} case, provided that only functions $b \in B$ are allowed as the argument.

6 Summary and Conclusions

We have shown the nature of MVC for G_{MX} for cases with different assumptions on the knowledge of the algorithm about the heuristics used and on the nature of the underlying graph. This enriches the theory on G_{MX} to more cases. We have also developed `MT`, which is equivalent to `fMM` but is simpler for our purposes.

Table 1 summarizes our findings. Each column represents a different case. For each case we provide the following information. The first row presents the structure of MVC. It is restrained for the I_{AD}/I_{CON} case and the $I_{AD\epsilon}/I_{CON\epsilon}$ case. For the I_{CON}/I_{CON} case, it is restrained for each equivalence class with a threshold given by a function $b \in B$. The next row gives the number of clusters of indistinguishable nodes in G_{MX}. For the I_{AD}/I_{CON} case and the $I_{AD\epsilon}/I_{CON\epsilon}$ case there is one cluster for each possible g-value. For the I_{CON}/I_{CON} case, we multiply this number by the number of equivalence classes which is $|\mathbb{H}|^2$ where $\mathbb{H}$ is the set of different h-values. The complexity column gives the best-known algorithm that calculates MVC given that G_{MX} and C^* are known. While an algorithm for the I_{AD}/I_{CON} case and the $I_{AD\epsilon}/I_{CON\epsilon}$ case were provided, for the I_{CON}/I_{CON} case one should resort to the known algorithm for finding minimal vertex covers in a bipartite graph (Papadimitriou and Steiglitz 1982).

Future work can further expand this theory to more cases such as the case where we add ϵ to the I_{CON}/I_{CON} case. In addition, an important line will be to develop such a theory to the case where we have a full front-to-front heuristic.

A Proofs for the Function b

We will first prove the first theorem that given a function $b \in B$, $\texttt{MT}_{CON}(b)$ will return an optimal path on any problem instance in I_{CON}. For that it is sufficient to show that there are always nodes from an optimal path in the open list with priority smaller or equal to C^*. We will prove this by showing that the priorities are non-decreasing along an optimal path and that any state on the optimal path has either a forward or a backward priority of C^* or less.

Lemma 15. *Let u, u' be two forward nodes on an optimal path such that u' is a direct descendant of u. Let $b : \mathbb{R}^+ \times \mathbb{R}^+ \rightarrow \mathbb{R}^+$ be a function. If $b \in B$ then $pr_F(u) \leq pr_F(u')$.*

Proof: If $g_F(u') \geq \underline{}(u')$ then $pr_F(u') = \infty$ and we are done. Otherwise, $g_F(u) \leq g_F(u') - h_C(u, u') \leq \underline{}(u') - h_C(u, u') \leq \underline{}(u)$ where the last inequality is directly from the definition of B. Now from consistency, $pr_F(u) = g_F(u) + h_F(u) \leq g_F(u') + h_F(u') = pr_F(u')$. $\square$

Lemma 16. *Let v, v' be two backward nodes on an optimal path such that v' is a direct descendant of v. Let $b : \mathbb{R}^+ \times \mathbb{R}^+ \rightarrow \mathbb{R}^+$ be a function. If $b \in B$ then $pr_B(v) \leq pr_B(v')$.*

Proof: It is enough to show that both terms in the max are non-decreasing. For the left term, from consistency $g_B(u) + h_B(u) \leq g_B(u') + h_B(u')$. As for the right term, $g_B(v) + b(v) \leq g_B(v') - h_C(v, v') + b(v) \leq g_B(v') + b(v')$ where the last inequality is directly from the definition of B. $\square$

Theorem 17. *Let $b : \mathbb{R}^+ \times \mathbb{R}^+ \rightarrow \mathbb{R}^+$ be a function. If $b \in B$ then $\texttt{MT}_{CON}(b)$ will return an optimal path on any problem instance in I_{CON}.*

Proof: Let s be a state on the optimal path. If $g_F(s) < b(s)$ then $pr_F(s) = f_F(s) \leq C^*$. Otherwise, $g_B(s) + b(s) = C^* - g_F(s) + b(s) \leq C^*$ and we know that $f_B(s) \leq C^*$, therefore $pr_B(s) \leq C^*$. Together with Lemmas 15 and 16 this means that there will always be on either open list a state from the optimal path with priority of C^*or less. $\square$

Next we will prove the second theorem, given a problem instance in I_{CON} we will construct a function b^* such that

$b^* \in B$ and $\mathtt{MT}_{CON}(b^*)$ will expand the minimal number of necessary expansions on that instance.

If MVC only contains forward nodes then $b^* = C^*$. Similarly if it only contains backward nodes, then $b^* = 0$. Below we will assume the non trivial case where nodes from both directions are inside MVC. We use $h_C(u, {h_F}', {h_B}')$ as short for $\max\{|{h_F}' - h_F(u)|, |{h_B}' - h_B(u)|\}$

Definition 6.

$$g_{F^{\text{I}}}({h_F}', {h_B}') = \max_{u \in (G_{\text{MX}_F} \cap \text{MVC})} \{g_F(u) - h_C(u, {h_F}', {h_B}')\}$$

$$g_{F^{\text{O}}}({h_F}', {h_B}') = \min_{u \in (G_{\text{MX}_F} \setminus \text{MVC})} \{g_F(u) + h_C(u, {h_F}', {h_B}')\}$$

$$g_{B^{\text{I}}}({h_F}', {h_B}') = \max_{v \in (G_{\text{MX}_B} \cap \text{MVC})} \{g_B(v) - h_C(u, {h_F}', {h_B}')\}$$

$$g_{B^{\text{O}}}({h_F}', {h_B}') = \min_{v \in (G_{\text{MX}_B} \setminus \text{MVC})} \{g_B(v) + h_C(u, {h_F}', {h_B}')\}$$

Let $b^* : \mathbb{R}^+ \times \mathbb{R}^+ \to \mathbb{R}^+$ be a function such that for every heuristics pair ${h_F}', {h_B}'$,

$$b^*({h_F}', {h_B}') = \frac{\max\{g_{F^{\text{I}}}({h_F}', {h_B}'), C^* - g_{B^{\text{O}}}({h_F}', {h_B}')\}}{2} + \frac{\min\{g_{F^{\text{O}}}({h_F}', {h_B}'), C^* - g_{B^{\text{I}}}({h_F}', {h_B}')\}}{2}$$

Lemma 18. *(helping inequalities)*
Let ${h_F}', {h_B}' \in \mathbb{R}^+$ be a pair of heuristic values.

1. $g_{F^{\text{I}}}({h_F}', {h_B}') < b^*({h_F}', {h_B}') \leq g_{F^{\text{O}}}({h_F}', {h_B}')$
2. $g_{F^{\text{I}}}({h_F}', {h_B}') < b^*({h_F}', {h_B}') < C^* - g_{B^{\text{I}}}({h_F}', {h_B}')$
3. $C^* - g_{B^{\text{O}}}({h_F}', {h_B}') \leq b^*({h_F}', {h_B}') \leq g_{F^{\text{O}}}({h_F}', {h_B}')$
4. $C^* - g_{B^{\text{O}}}({h_F}', {h_B}') \leq b^*({h_F}', {h_B}') < C^* - g_{B^{\text{I}}}({h_F}', {h_B}')$

Proof: Since t_F^* is defined as the mean of the maximum of the left terms and the minimum of the right terms, it is sufficient to prove each inequality only between the left and the right terms. □

1. $g_{F^{\text{I}}}({h_F}', {h_B}') < g_{F^{\text{O}}}({h_F}', {h_B}')$
   Proof: From Definition 6 there must exist two forward nodes $u \in$ MVC and $u' \notin$ MVC such that $g_F(u) - h_C(u, {h_F}', {h_B}') = g_{F^{\text{I}}}({h_F}', {h_B}')$ and $g_F(u') + h_C(u', {h_F}', {h_B}') = g_{F^{\text{O}}}({h_F}', {h_B}')$. According to Lemma 3 there exists a backward node $v \notin$ MVC such that $g_F(u) + g_B(v) + h_C(u, v) < C^*$. We know that the pair (u', v) is not a *must-expand pair* since both are not in MVC, therefore $g_F(u') + g_B(v) + h_C(u', v) \geq C^*$. Now we have $g_{F^{\text{I}}}({h_F}', {h_B}') = g_F(u) - h_C(u, {h_F}', {h_B}') < C^* - h_C(u, v) - g_B(v) - h_C(u, {h_F}', {h_B}') \leq g_F(u') + h_C(u', v) - h_C(u, v) - h_C(u, {h_F}', {h_B}') \leq g_F(u') + h_C(u', {h_F}', {h_B}') = g_{F^{\text{O}}}({h_F}', {h_B}')$ where the last inequality is Lemma 8 used multiple times. □
2. $g_{F^{\text{I}}}({h_F}', {h_B}') < C^* - g_{B^{\text{I}}}({h_F}', {h_B}')$
   Proof: From Definition 6 there must exist a forward node $u \in$ MVC and a backward node $v \in$ MVC such that $g_F(u) - h_C(u, {h_F}', {h_B}') = g_{F^{\text{I}}}({h_F}', {h_B}')$ and $g_B(v) - h_C(v, {h_F}', {h_B}') = g_{B^{\text{I}}}({h_F}', {h_B}')$. According to Lemma 3 there exists a backward node $v' \notin$ MVC such that $g_F(u) + g_B(v') + h_C(u, v') < C^*$. Similarly there exists a forward node $u' \notin$ MVC such that $g_F(u') + g_B(v) + h_C(u', v) < C^*$. We know that the pair (u', v') is not a *must-expand pair* since both are not in MVC, therefore $g_F(u') + g_B(v') + h_C(u', v') \geq C^*$. Now we have $g_{F^{\text{I}}}({h_F}', {h_B}') = g_F(u) - h_C(u, {h_F}', {h_B}') < C^* - h_C(u, v') - g_B(v') - h_C(u, {h_F}', {h_B}') \leq g_F(u') + h_C(u', v') - h_C(u, v') - h_C(u, {h_F}', {h_B}') < C^* - g_B(v) - h_C(u', v) + h_C(u', v') - h_C(u, v') - h_C(u, {h_F}', {h_B}') \leq C^* - g_B(v) + h_C(v, {h_F}', {h_B}') = C^* - g_{B^{\text{I}}}({h_F}', {h_B}')$ where the last inequality is Lemma 8 used multiple times. □
3. $C^* - g_{B^{\text{O}}}({h_F}', {h_B}') \leq g_{F^{\text{O}}}({h_F}', {h_B}')$
   Proof: From Definition 6 there must exist a forward node $u \notin$ MVC and a backward node $v \notin$ MVC such that $g_F(u) + h_C(u, {h_F}', {h_B}') = g_{F^{\text{O}}}({h_F}', {h_B}')$ and $g_B(v) + h_C(v, {h_F}', {h_B}') = g_{B^{\text{O}}}({h_F}', {h_B}')$. We know that (u, v) is not a must-expand pair since both are not in MVC, therefore $g_F(u) + g_B(v) + h_C(u, v) \geq C^*$. Now we have $C^* - g_{B^{\text{O}}}({h_F}', {h_B}') = C^* - g_B(v) - h_C(v, {h_F}', {h_B}') \leq g_F(u) + h_C(u, v) - h_C(v, {h_F}', {h_B}') \leq g_F(u) + h_C(u, {h_F}', {h_B}') = g_{F^{\text{I}}}({h_F}', {h_B}')$ where the last inequality is Lemma 8. □
4. $C^* - g_{B^{\text{O}}}({h_F}', {h_B}') < C^* - g_{B^{\text{I}}}({h_F}', {h_B}')$
   Proof: This is equivalent to proving that $g_{B^{\text{I}}}({h_F}', {h_B}') < g_{B^{\text{O}}}({h_F}', {h_B}')$ which is identical to inequality (1) except for the direction, so the proof is similar as well. □

Lemma 19. *For a forward node u, $u \in$ MVC iff $g_F(u) < b^*(h_F(u), h_B(u))$.*

Proof: If $u \in$ MVC, then directly from Definition 6 we know that $g_F(u) \leq g_{F^{\text{I}}}$ and from Lemma 18 $g_F(u) < t_F^*$. If $u \notin$ MVC then directly from Definition 6 we know that $g_F(u) \geq g_{F^{\text{O}}}$ and from Lemma 18 $g_F(u) \geq t_F^*$. □

Lemma 20. *Let $b_1, b_2 : \mathbb{R}^+ \times \mathbb{R}^+ \to \mathbb{R}^+$ be two functions. If $b_1, b_2 \in B$ then $b_{\max} = \max\{b_1, b_2\} \in B$ and $b_{\min} = \min\{b_1, b_2\} \in B$.*

Proof: Let $x_1, y_1, x_2, y_2 \in \mathbb{R}^+$, we want to show that $|b_{\max}(x_1, y_1) - b_{\max}(x_2, y_2)| \leq \max\{|x_1 - x_2|, |y_1 - y_2|\}$. If $b_{\max}(x_1, y_1) = b_1(x_1, y_1)$ and $b_{\max}(x_2, y_2) = b_1(x_2, y_1)$ then we are done. Otherwise, assume w.l.o.g that $b_{\max}(x_1, y_1) = b_1(x_1, y_1)$, $b_{\max}(x_2, y_2) = b_2(x_2, y_2)$ (otherwise we can switch between b_1 and b_2) and that $b_{\max}(x_1, y_1) \geq b_{\max}(x_2, y_2)$ (otherwise we can switch between $(x1, y1)$ and $(x2, y2)$). Now from our assumptions, $|b_{\max}(x_1, y_1) - b_{\max}(x_2, y_2)| = b_1(x_1, y_1) - b_2(x_2, y_2) \leq b_1(x_1, y_1) - b_1(x_2, y_2) \leq \max\{|x_1 - x_2|, |y_1 - y_2|\}$. The proof for $b_{\min}$ is similar. □

Lemma 21. *Let $b_1, b_2 : \mathbb{R}^+ \times \mathbb{R}^+ \to \mathbb{R}^+$ be two functions. If $b_1, b_2 \in B$ then $b_{mean} = \frac{b_1}{2} + \frac{b_2}{2} \in B$.*

Proof: Let $x_1, y_1, x_2, y_2 \in \mathbb{R}^+$, we want to show that $|b_{\text{mean}}(x_1, y_1) - b_{\text{mean}}(x_2, y_2)| \leq \max\{|x_1 - x_2|, |y_1 - y_2|\}$. Using the triangle inequality, $|b_{\text{mean}}(x_1, y_1) - b_{\text{mean}}(x_2, y_2)| = \left|\frac{b_1(x_1,y_1)}{2} + \frac{b_2(x_1,y_1)}{2} - \frac{b_1(x_2,y_2)}{2} - \frac{b_2(x_2,y_2)}{2}\right| \leq$

$\left|\frac{b_1(x_1,y_1)-b_1(x_2,y_2)}{2}\right| + \left|\frac{b_2(x_1,y_1)-b_2(x_2,y_2)}{2}\right| \leq 2 \times \frac{1}{2}\max\{|x_1-x_2|, |y_1-y_2|\} = \max\{|x_1-x_2|, |y_1-y_2|\}$ □

Given a forward node u, it is trivial that $g_F(u) \pm h_C(u, {h_F}', {h_B}') \in B$ and similarly for backward nodes. Therefore, b^* is constructed from functions in B using only three operators: max, min and mean. According to Lemmas 20 and 21, B is closed under these operators, therefore $b^* \in B$.

Theorem 22. *Given a problem instance in I_{CON} there exists $b \in B$ such that $\texttt{MT}_{CON}(b)$ will expand the minimal number of necessary expansions on that instance.*

Proof: According to Lemmas 20 and 21, b^* as defined above is in B. According to Lemma 19, $\texttt{MT}_{CON}(b^*)$ will only expand the nodes in MVC hence it will expand the minimal number of necessary expansions on that instance.

B Acknowledgments

The work was partially supported by the Israel Science Foundation (ISF) under grant #844/17 to Ariel Felner. This research has also been partially supported by the HUJI Cyber Security Research Center in conjunction with the Israel National Cyber Directorate (INCD) in the Prime Minister's Office.

References

Chen, J.; Holte, R. C.; Zilles, S.; and Sturtevant, N. R. 2017. Front-to-end bidirectional heuristic search with near-optimal node expansions. In *Proceedings of IJCAI*.

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*.

Holte, R. C.; Felner, A.; Sharon, G.; Sturtevant, N. R.; and Chen, J. 2017. Bidirectional search that is guaranteed to meet in the middle. *Artificial Intelligence Journal (AIJ), In press*.

Kaindl, H., and Kainz, G. 1997. Bidirectional heuristic search reconsidered. *J. Artif. Intell. Res.* 7:283–317.

Papadimitriou, C. H., and Steiglitz, K. 1982. *Combinatorial optimization: algorithms and complexity*. Courier Corporation.

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.