Extended Abstract: An Improved Priority Function for Bidirectional Heuristic Search

Guni Sharon  Robert C. Holte  Ariel Felner  Nathan R. Sturtevant
Computer Science  Computing Science Dept.  Information Systems Engineering  Computer science
University of Texas at Austin  University of Alberta  Ben-Gurion University  University of Denver
Austin, Texas, U.S.A.  Edmonton, Canada T6G 2E8  Beer-sheva, Israel  Denver, Colorado, USA
(gunisharon@gmail.com)  (rholte@ualberta.ca)  (felner@bgu.ac.il)  (sturtevant@cs.du.edu)

1 Introduction

Bidirectional search algorithms interleave a search forward from the start state (start) and a search backward (i.e. using reverse operators) from the goal state (goal). We say that the two searches “meet in the middle” if neither search expands a node whose g-value (in the given direction) exceeds C*/2, where C* is the cost of an optimal solution. The only bidirectional heuristic search algorithm that is guaranteed to meet in the middle under all circumstances is the recently introduced M&M algorithm (Holte et al. 2016). The feature of M&M that provides this guarantee is its unique priority function for nodes on its open lists.

In this short note we present MME, which enhances M&M’s priority function and is expected to expand fewer nodes than M&M under most circumstances. We sketch a proof of MME’s correctness, describe conditions under which MME will expand fewer nodes than M&M and vice versa, and experimentally compare M&M and MME on the 10-Pancake problem.

2 The MME Algorithm

MME runs an A*-like search in both directions, so we use the usual notation—g, h, f, Open, etc.—but have separate copies of these variables for the two search directions, with a subscript (F or B) indicating the direction:

Forward search: fF, gF, hF, OpenF, ClosedF, etc.
Backward search: fB, gB, hB, OpenB, ClosedB, etc.

Node Priority: MME chooses to expand a node n from either OpenF or OpenB with minimum “priority”, which is defined for OpenF (and analogously for prB(n)) as:

\[ prF(n) = \max(fF(n), 2gF(n)) \]  

Stopping condition: MME keeps track of the cheapest path from start to goal that it has seen so far, recording its cost in the variable U (U = \( \infty \) initially). MME detects that a new path from start to goal has been found by checking if a node generated in one search direction is present in the open list of the other search direction, and updates U if necessary. MME terminates as soon as U \( \leq C \), where C is the smallest priority of any node on either open list.\(^1\)

\(^1\)MME has additional stopping conditions so that it can terminate sooner, but this stopping condition suffices for the proofs below.

MME has three important properties:

(P1) MME’s forward and backward searches meet in the middle, i.e. neither search expands a state whose distance from the search’s origin (start for forward search, goal for backward search) is larger than C*/2.

(P2) MME never expands a node whose f-value exceeds C*.

(P3) If there exists a path from start to goal M&M returns C*.

We wish to show that MME has these properties as well. To do that, we will first review the proof for M&M and then show that the same proof applies to MME. The proof for M&M is based on the following three lemmas.

L1: Let \( d(x, y) \) be the cost of a least-cost path from state x to state y. If \( d(\text{start}, s) > C*/2 \), then \( prF(s) > C* \), and if \( d(s, \text{goal}) > C*/2 \), then \( prB(s) > C* \).

Proof (for the forward direction): \( prF(s) \geq 2gF(s) \geq 2d(\text{start}, s) \). If \( d(\text{start}, s) > C*/2 \) then \( prF(s) > C* \). □

We say that a path from start to goal has been “found” if one or more nodes on the path have been opened in both directions.

L2: If \( P \) is an optimal path that has not been found, there will exist a node \( n \in P \) such that \( n \in \text{OpenF} \) with \( prF(n) \leq C* \) or \( n \in \text{OpenB} \) with \( prB(n) \leq C* \).

Proof: Throughout M&M’s execution there will be a node, \( nF \), from \( P \) in \( \text{OpenF} \) with \( gF(nF) = \min \{d(\text{start}, n), \text{goal}\} \), and a node, \( nB \), from \( P \) in \( \text{OpenB} \) with \( gB(nB) = d(nB, \text{goal}) \). Since \( P \) has not yet been found there must exist a gap between \( nF \) and \( nB \), i.e. one or more edges from \( P \) that connect \( nF \) to \( nB \) that M&M has not traversed. This situation is depicted in Figure 1 (left), where the dashed line between \( nF \) and \( nB \) is the gap consisting of one or more edges. In this situation, either \( gF(nF) \leq \min \{d(\text{start}, n), \text{goal}\} \leq \min \{prF(n), prB(n)\} \leq \min \{C*, \text{cost}(P)\} \) or both, where \( \text{cost}(P) = C* \) is the sum of the costs on \( P \)’s edges. Therefore \( prF(nF) \leq \min \{d(\text{start}, n), \text{goal}\} \leq \min \{prF(n), prB(n)\} \leq \min \{C*, \text{cost}(P)\} \) or both. □

L3: \( U > C* \) until the first optimal path from start to goal is found, at which point \( U = C* \). This is a direct consequence of the process by which U is updated.

We now sketch the proof that MME has properties P1–P3. L2 and L3 together ensure that MME will not terminate before an optimal path is found (L2 implies that \( C \leq C* \) until all optimal paths have been found and L3 says \( U > C* \) until
the first optimal path is found). P3 follows because $U = C^*$ once an optimal path is found (L3).

L2 and L1 together ensure that MM will find an optimal path before it expands any node in the forward direction with $pr_F(n) > C^*$ or any node in the backward direction with $pr_B(n) > C^*$. Together with L3 this implies that MM will terminate before it expands any node in the forward direction with $f_F(n) > C^*$ or $d(start, s) > C^*/2$, or any node in backward direction with $f_B(n) > C^*$ or $d(s, goal) > C^*/2$, thus proving P1 and P2.

3 MM

$ε_F(n) (ε_B(n))$ is the cost of the cheapest forward (reverse) operator applicable to $n$. MM is identical to MM except for a small change in how an open node’s priority is defined. In MM the priority of $n ∈ Open_F$ is

$$ pr_F(n) = \max(f_F(n), 2g_F(n) + ε_F(n)) $$

$pr_B(n)$ is defined analogously. We now prove that MM has properties P1–P3 by showing that facts L1–L3 hold for MM.

L1 is still true since $pr_F(s) ≥ pr_F(s)$ and $pr_B(s) ≥ pr_B(s). L3$ is still true because it is not affected by the definition of a state’s priority. To see that L2 is still true, note that $d(n_F, n_B)$ is the cost of the gap illustrated in Figure 1 (left), i.e. $C^* = g_F(n_F) + d(n_F, n_B) + g_B(n_B)$. Hence, at least one of $g_F(n_F)$ and $g_B(n_B)$ must be less than or equal to $(C^* - d(n_F, n_B))/2$. The exact value of $d(n_F, n_B)$ is not known, but it always holds that $ε_F(n_F) ≤ d(n_F, n_B)$. Similarly, $ε_B(n_B) ≤ d(n_F, n_B)$. Therefore, either $g_F(n_F) ≤ (C^* - ε_F(n_F))/2$ or $g_B(n_B) ≤ (C^* - ε_B(n_B))/2$. If $g_B(n_F) ≤ (C^* - ε_F(n_F))/2$ then $2g_B(n_F) + ε_F(n_F) ≤ C^*$. Similar reasoning applies for $g_B(n_B)$. L2 follows because at least one of these must hold.

MM vs. MM, $ε_F(n) ≥ 0$, $pr_F(n) ≥ pr_F(n)$, therefore MM is expected to outperform MM for the same reason that A*’s performance is expected to improve with a better heuristic function. But just as a better heuristic may cause A* to expand more nodes (Holte 2010), so too MM may expand more nodes than MM.

Figure 1 (center) illustrates why MM will often expand fewer nodes than MM. Here $g_F(X) = 2$, $h_F(X) = 1$ and $ε_F(X) = 2$. Therefore, $pr_F(X) = 4$ while $pr_F(X) = 4 + ε_F(X) = 6$. After MM expands start, Open_F includes three nodes, X, A, and goal with priorities 4, 7, and 10, respectively. At this point $U = 5$. Now MM expands goal in the backward direction ($pr_B(goal) = 3$). At this point X is the only open node with priority less than $U = 5$ so it is expanded and MM halts. MM also expands start in the forward direction and goal in the backward direction, but it can halt at that point, without expanding X, because $pr_F(X) = 6 > U$.

### Table 1: 10-pancake: nodes expanded for $C^* = 11$

<table>
<thead>
<tr>
<th></th>
<th>h = 0</th>
<th>GAP-3</th>
<th>GAP-2</th>
<th>GAP-1</th>
<th>GAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>A*</td>
<td>2,801,751</td>
<td>302,363</td>
<td>80,239</td>
<td>12,629</td>
<td>318</td>
</tr>
<tr>
<td>MM</td>
<td>9,449</td>
<td>37,403</td>
<td>29,925</td>
<td>8,883</td>
<td>478</td>
</tr>
<tr>
<td>MM #3</td>
<td>9,449</td>
<td>8,681</td>
<td>8,297</td>
<td>3,751</td>
<td>342</td>
</tr>
</tbody>
</table>

Figure 1 (right) is an example where MM expands more nodes than MM when both use all of MM’s stopping conditions (Holte et al. 2016). Both algorithms begin by expanding start (forward) and goal (backward). Node A will not be expanded in the forward direction by either algorithm because $g_F(A) = 8 > 6 = C^*/2$ and both algorithms will halt as soon as A is expanded in the backward direction (when that happens $U = 12 ≤ gmin_F + gmin_B + ϵ = 18$ (gmin_X is the minimum g-value in Open_X (X ∈ {B, F}) and $ε$ is the smallest edge cost in the space). For both algorithms $pr_F(X) = f_F(X) = 9$. For MM, $pr_B(A) = 8$ so MM will expand it before X and then halt. For MM, $pr_B(A) = 12$ so MM will expand X before A.

### 4 Experimental results

Table 1 shows the average number of nodes expanded over 10 instances, all with $C^* = 11$, on the 10-pancake problem. The constant time per node was very similar for all algorithms and thus time is not reported. To examine the effect of the heuristic’s accuracy on the relative performance of the algorithms, we used heuristics of varying accuracy. We used the GAP heuristic (Helmert 2010) and created less accurate heuristics from it, referred to as GAP-X, by not counting the gaps involving any of the X smallest pancakes. For example, GAP-2 does not count the gaps involving pancakes 0 or 1.

A* is the best algorithm for the very accurate GAP heuristic. As explained by Holte et al. (2016), for weaker heuristics MM and MM #3 outperform A*. MM is always better than MM, by a factor of up to 4. Holte et al. (2016) describe an “anomaly” that MM without a heuristic (first column) can outperform MM with weak heuristics. This can be seen for MM with GAP-2 and GAP-3. Although this anomaly is theoretically possible for MM, it does not occur in this experiment.

### References