Proceedings of the Ninth International
Symposium on Combinatorial Search (SoCS 2016)

Extended Abstract: An Improved Priority Function for Bidirectional Heuristic Search

Guni Sharon
Computer Science
University of Texas at Austin
Austin, Texas, U.S.A
(gunisharon@gmail.com)

Robert C. Holte
Computing Science Dept.
University of Alberta
Edmonton, Canada T6G 2E8
(rholte@ualberta.ca)

Ariel Felner
Information Systems Engineering
Ben-Gurion University
Beer-sheva, Israel
(felner@bgu.ac.il)

Nathan R. Sturtevant
Computer science
University of Denver
Denver, Colorado, USA
(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 MM algorithm (Holte et al. 2016). The feature of MM that provides this guarantee is its unique priority functions for nodes on its open lists.

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

2 The MM Algorithm

MM 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: f_F, g_F, h_F, $Open_F$, $Closed_F$, etc.
Backward search: f_B, g_B, h_B, $Open_B$, $Closed_B$, etc.

Node Priority: MM chooses to expand a node n from either $Open_F$ or $Open_B$ with minimum "priority", which is defined for $Open_F$ (and analogously for $pr_B(n)$) as:

$$pr_F(n) = \max(f_F(n), 2g_F(n)) \quad (1)$$

Stopping condition: MM 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). MM 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. MM terminates as soon as $U \leq C$, where C is the smallest priority of any node on either open list.[1]

[1] MM has additional stopping conditions so that it can terminate sooner, but this stopping condition suffices for the proofs below.

MM has three important properties:

- **(P1)** MM'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)** MM never expands a node whose f-value exceeds C^*.
- **(P3)** If there exists a path from $start$ to $goal$ MM returns C^*.

We wish to show that MMϵ has these properties as well. To do that, we will first review the proof for MM and then show that the same proof applies to MMϵ. The proof for MM 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 $d(start, s) > C^*/2$, then $pr_F(s) > C^*$, and if $d(s, goal) > C^*/2$, then $pr_B(s) > C^*$.

Proof (for the forward direction): $pr_F(s) \geq 2g_F(s) \geq 2d(start, s)$. If $d(start, s) > C^*/2$ then $pr_F(s) > C^*$. □

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 $n \in Open_F$ with $pr_F(n) \leq C^*$ or $n \in Open_B$ with $pr_B(n) \leq C^*$.

Proof: Throughout MM's execution there will be a node, n_F, from P in $Open_F$ with $g_F(n_F) = d(start, n_F)$, and a node, n_B, from P in $Open_B$ with $g_B(n_B) = d(n_B, goal)$. Since P has not yet been found there must exist a gap between n_F and n_B, i.e. one or more edges from P that connect n_F to n_B that MM has not traversed. This situation is depicted in Figure 1 (left), where the dashed line between n_F and n_B is the gap consisting of one or more edges. In this situation, either $g_F(n_F) \leq cost(P)/2$ or $g_B(n_B) \leq cost(P)/2$ (or both), where $cost(P) = C^*$ is the sum of the costs on P's edges. Therefore $pr_F(n_F) \leq cost(P) = C^*$ or $pr_B(n_B) \leq cost(P) = C^*$ (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 MM has properties P1–P3. L2 and L3 together ensure that MM 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

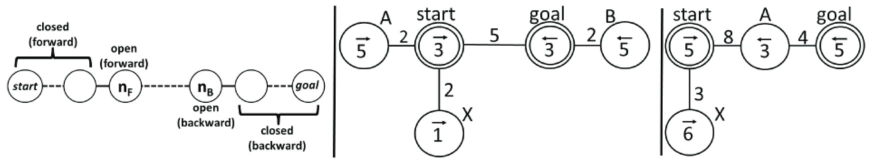

Figure 1: *Left:* The gap on an optimal path. *Center:* MMϵ expands fewer nodes. *Right:* MM expands fewer nodes.

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 MMϵ

$\epsilon_F(n)$ ($\epsilon_B(n)$) is the cost of the cheapest forward (reverse) operator applicable to n. MMϵ is identical to MM except for a small change in how an open node's priority is defined. In MMϵ the priority of $n \in Open_F$ is

$$pr_F^{\epsilon}(n) = \max(f_F(n), 2g_F(n) + \epsilon_F(n)) \qquad (2)$$

$pr_B^{\epsilon}(n)$ is defined analogously. We now prove that MMϵ has properties P1–P3 by showing that facts L1–L3 hold for MMϵ.

L1 is still true since $pr_F^{\epsilon}(s) \geq pr_F(s)$ and $pr_B^{\epsilon}(s) \geq pr_B(s)$. 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 $\epsilon_F(n_F) \leq d(n_F, n_B)$. Similarly, $\epsilon_B(n_B) \leq d(n_F, n_B)$. Therefore, either $g_F(n_F) \leq (C^* - \epsilon_F(n_F))/2$ or $g_B(n_B) \leq (C^* - \epsilon_B(n_B))/2$. If $g_F(n_F) \leq (C^* - \epsilon_F(n_F))/2$ then $2g_F(n_F) + \epsilon_F(n_F) \leq C^*$. Similar reasoning applies for $g_B(n_B)$. L2 follows because at least one of these must hold.

MM vs. MM$_\epsilon$. $\epsilon_F(n) \geq 0, pr_F^{\epsilon}(n) \geq pr_F(n)$, therefore MMϵ 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 MMϵ may expand more nodes than MM.

Figure 1 (center) illustrates why MMϵ will often expand fewer nodes than MM. Here $g_F(X) = 2$, $h_F(X) = 1$ and $\epsilon_F(X) = 2$. Therefore, $pr_F(X) = 4$ while $pr_F^{\epsilon}(X) = 4 + \epsilon_F(X) = 6$. 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. MMϵ also expands $start$ in the forward direction and $goal$ in the backward direction, but it can halt at that point, without expanding X, because $pr_F^{\epsilon}(X) = 6 > U$.

	$h \equiv 0$	GAP-3	GAP-2	GAP-1	GAP
A*	2,801,751	302,363	80,239	12,629	**318**
MM	**9,449**	37,403	29,925	8,883	478
MMϵ	**9,449**	**8,681**	**8,297**	**3,751**	342

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

Figure 1 (right) is an example where MMϵ 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 $U = 12 \leq gmin_F + gmin_B + \epsilon = 18$ ($gmin_X$ is the minimum g-value in $Open_X$ ($X \in \{B, F\}$) 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 MMϵ, $pr_B(A) = 12$ so MM will expand X before A.

4 Experimental results

Table 1 shows the average number of nodes expanded over 30 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 MMϵ outperform A*. MMϵ 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 MMϵ, it does not occur in this experiment.

References

Helmert, M. 2010. Landmark heuristics for the pancake problem. In *Proc. 3rd Annual Symposium on Combinatorial Search, (SoCS)*.

Holte, R.; Felner, A.; Sharon, G.; and Sturtevant, N. 2016. Bidirectional search that is guaranteed to meet in the middle. In *AAAI*.

Holte, R. C. 2010. Common misconceptions concerning heuristic search. In *SoCS*.