

Improved Multi-Heuristic A* for Searching with Uncalibrated Heuristics

Venkatraman Narayanan

Carnegie Mellon University
Pittsburgh, PA, USA
venkatraman@cs.cmu.edu

Sandip Aine

Indraprastha Institute of Information
Technology, New Delhi, India
sandip@iiitd.ac.in

Maxim Likhachev

Carnegie Mellon University
Pittsburgh, PA, USA
maxim@cs.cmu.edu

Abstract

Recently, several researchers have brought forth the benefits of searching with multiple (and possibly inadmissible) heuristics, arguing how different heuristics could be independently useful in different parts of the state space. However, algorithms that use inadmissible heuristics in the traditional best-first sense, such as the recently developed Multi-Heuristic A* (MHA*), are subject to a crippling *calibration problem*: they prioritize nodes for expansion by *additively* combining the cost-to-come and the inadmissible heuristics even if those heuristics have no connection with the cost-to-go (e.g., the heuristics are uncalibrated). For instance, if the inadmissible heuristic were an order of magnitude greater than the perfect heuristic, an algorithm like MHA* would simply reduce to a weighted A* search with one consistent heuristic. In this work, we introduce a general multi-heuristic search framework that solves the calibration problem and as a result a) facilitates the effective use of multiple uncalibrated inadmissible heuristics, and b) provides significantly better performance than MHA* whenever tighter sub-optimality bounds on solution quality are desired. Experimental evaluations on a complex full-body robotics motion planning problem and large sliding tile puzzles demonstrate the benefits of our framework.

Introduction

The quality of the heuristic makes or breaks informed search algorithms such as A* (Hart, Nilsson, and Raphael 1968). An admissible heuristic guarantees optimality whereas an informative (goal-directed) heuristic leads to faster solutions. Unfortunately, designing heuristics that are both informative *and* admissible is often challenging. To address this, several algorithms have been developed over the years that trade off optimality for speed by using heuristics that are more goal-directed but inadmissible.

Inadmissible heuristics used in literature typically fall in one of the following classes: (a) an inflated admissible heuristic or a weighted sum of admissible heuristics, (b) an inadmissible estimate/approximation of an admissible heuristic such as h^+ (Hoffmann 2005), (c) an estimate of the distance (number of edges) to get to the goal and, (d) an arbitrary function that has nothing to do with the cost being

optimized (for example, a function of state-dependent features or some domain-specific inadmissible heuristic). Most informed search algorithms typically treat the inadmissible heuristics in the same way as the admissible ones by computing a priority $g(s) + h(s)$ to determine which state to expand. Computing the priority in this fashion poses unfortunate problems especially for cases (c) and (d) as we are mixing two fundamentally different quantities— g , which is an estimate of the cost-to-come and h , the heuristic which might have nothing to do with the cost-to-go. We term this the *calibration problem* since g and h could be operating on different units or scales.

The following example better illustrates the calibration problem. Consider a robotics motion planning problem where a dual-arm personal robot needs to move from one room to another, and we want to minimize the distance traveled by the robot. Each state in the graph represents the configuration of the robot: its position in the world and the configuration of its arms. Suppose that the robot starts out with its arm spread out and that the search uses an admissible heuristic computed as the Euclidean distance from the current (x, y) location to the goal (x, y) location of the robot base. While the search could start out well by expanding states towards the goal it might get stuck at a doorway because its arms are spread out, thereby unable to generate successor states that pass through the doorway. A typical heuristic one might try to get around this problem is to say, for states near the door, prefer expanding those states where the arms are closer to being ‘tucked-in’ as they are more likely to get the robot through the doorway. Such a heuristic has no connection to the actual cost being minimized (the distance traveled) and therefore it means very little to compute quantities such as $g(s) + h(s)$. This calls for an algorithm that can handle *uncalibrated heuristics* in a non-additive fashion and yet provide good quality solutions.

Recent work (Isto 1996; Röger and Helmert 2010; Aine et al. 2014) furthers the utility of inadmissible heuristics by convincingly arguing how *multiple* inadmissible heuristics could improve search performance. The common argument presented in these works is that each individual heuristic might be useful in some part of the state space, and therefore it might be more beneficial to use them in an independent manner as opposed to combining them into one heuristic. To build upon the previous example, the ‘tuck-arm’ heuris-

tic might be useful when expanding states near the door, whereas an ‘untuck-arm’ heuristic might be useful near the goal state, say, where the robot needs to pick up an object. Clearly, typical additive or max-like combinations of these heuristics are not meaningful here—the search would prefer expanding states where the robot arm is halfway between tucked and untucked. On the other hand, the search would benefit if they were used independently in different parts of the state-space, such as one near the doorway, and the other near the goal region.

In this work, we present a general multi-heuristic search framework that is similar in spirit to Multi-Heuristic A* (MHA*) (Aine et al. 2014), while making several significant improvements including fixing the calibration problem that could degenerate MHA* to weighted A* with one consistent heuristic. We summarize the contributions and improvements over MHA* below:

1. A general multi-heuristic search framework, Improved MHA*, which solves the calibration problem and thereby permits the efficient use of multiple uncalibrated (and possibly inadmissible) heuristics. This allows for a complete decoupling between the design of graph costs and heuristics.
2. Three instantiations of the general framework with discussion on the pros and cons of each. We also show that the original MHA* algorithm can be obtained as a variation of our more general framework.
3. Theoretical guarantees on completeness, bounds on sub-optimality of the solution and bounds on number of re-expansions of any state for all Improved MHA* variants, when one consistent heuristic is available.

We evaluate the different instantiations of our framework on two complex planning problems: the first, a real-world robotics 11-DoF full-body motion planning problem for the PR2 robot and the second, a large sliding tile puzzle problem. Our experiments show how the proposed framework is able to utilize several uncalibrated heuristics seamlessly for finding solutions.

Related Work

Inadmissible Heuristics

The use of inadmissible heuristics to trade off optimality for speed has been well studied in heuristic search literature. Most notable amongst algorithms that use inadmissible heuristics is weighted A* or wA* (Pohl 1970), which constructs an inadmissible heuristic by inflating an admissible one. While inflated heuristics typically do help find solutions faster because of their goal-directedness and help guarantee w -optimality of the solution, they are limited in the sense that they can only be constructed from admissible heuristics. This rules out a large class of inadmissible heuristics that could potentially help find a solution faster.

Inadmissible heuristics used in classical planning usually try to approximate an admissible heuristic such as h^+ (Hoffmann 2005). Some examples of these approximations include h^{add} , h^{FF} , h^{1st} , h^{LAMA} etc. (Betz and Helmert 2009). Since these are inadmissible estimates, they can be larger

than h^+ by an arbitrarily large multiplicative factor (Betz and Helmert 2009) or in other words, uncalibrated with the actual cost-to-go. Consequently, using these uncalibrated heuristics in the usual best-first manner by computing priorities such as $g(s) + h(s)$ could be inefficient as two out-of-scale quantities are additively combined. Note that the calibration problem exists only for best-first methods, i.e., when we care about solution quality, and not when using a greedy search algorithm. While many researchers opt for greedy searches to find solutions faster, we target problems where optimizing solution cost is also important.

Algorithms based on A_ϵ^* (Pearl and Kim 1982) use an inadmissible heuristic to speed up search, albeit in a different fashion. They operate by constructing a prefix of the OPEN list called the FOCAL list and use a second heuristic (typically an inadmissible estimate of the number of ‘expansions-to-go’ from a given state) to decide which state to expand from the FOCAL list. Explicit Estimation Search or EES and its variants (Thayer and Ruml 2011; Hatem and Ruml 2014) use a similar idea but employ a secondary inadmissible heuristic to remove bias in the construction of the FOCAL list. The success of these methods, though, predominantly relies on getting a good estimate of the number of ‘expansions-to-go’. Further, these methods require states to be re-expanded arbitrarily many times for guaranteeing desired sub-optimality bounds (Ebendt and Drechsler 2009) on the solution cost. Nevertheless, our work builds upon some of the ideas in these algorithms.

Multi-Heuristic Search

As noted in the introduction, searching with multiple heuristics in an independent manner can be beneficial as it allows users to design several heuristics that can be useful in different parts of the state space. The Multi-Heuristic A* (MHA*)¹ algorithm (Aine et al. 2014) operates on this idea and uses multiple inadmissible heuristics simultaneously to explore the search space. Despite allowing inadmissible heuristics, MHA* can provide bounds on solution quality and number of re-expansions of any state through the use of a single consistent heuristic. While MHA* allows for the use of a rich class of potentially informative inadmissible heuristics (unlike weighted A* or A_ϵ^* which constrain the inadmissibility in some sense), it suffers from a serious calibration problem: MHA* cycles through each available inadmissible heuristic and decides whether to use an inadmissible heuristic or the consistent heuristic based on an *anchor* condition. However, this anchor condition involves additively combining the cost-to-come g and the inadmissible heuristic, leading to the calibration problem explained earlier. As a result, it could happen that the anchor condition fails every time, thereby basically reducing to running weighted A* with a consistent heuristic.

A multi-heuristic greedy best-first search is described in (Röger and Helmert 2010) for satisficing planning. This operates in a manner similar to MHA* by maintaining several OPEN lists, each sorted based on a different heuristic.

¹Throughout the paper, we will use MHA* to refer to the Shared MHA* (SMHA*) variant (Aine et al. 2014)

States generated by an expansion from one OPEN list are shared with all the other OPEN lists so that every heuristic can independently look at different parts of the state space. While this algorithm does not suffer from the calibration problem (since there is no notion of cost-to-come), the solution quality could be arbitrarily bad and it cannot find solutions that are within some user-defined sub-optimality bound. A similar multi-heuristic search algorithm was used for robotics motion planning (Isto 1996), but is also unable to guarantee any bounds on solution quality.

In this work, we address the calibration problem that MHA* suffers from while preserving its key benefit of exploring the state space with multiple heuristics. The proposed multi-heuristic search framework, Improved MHA*, has all the properties of MHA* while providing stronger guarantees on efficiency.

Background

In this section we define the notation and terminology used through the rest of the paper and provide an overview of MHA* (Aine et al. 2014), on which our framework is developed.

Notation and Terminology

We assume the given planning problem can be represented as a graph-search problem. Let S denote the finite set of states of the planning domain. For an edge between s and s' , $c(s, s')$ denotes the cost of the edge, and if there is no such edge, then $c(s, s') = \infty$. The successor function $\text{SUCC}(s) := \{s' \in S \mid c(s, s') \neq \infty\}$, denotes the set of all reachable successors of s . An optimal path from state s to s' has cost $c^*(s, s')$ and the optimal path from s_{start} to s has cost $g^*(s)$.

Let $g(s)$ denote the current best path cost from s_{start} to s and $h(s)$ denote the heuristic for s , typically an estimate of the best path cost from s to s_{goal} . A heuristic is *admissible* if it never overestimates the best path cost to s_{goal} and *consistent* if it satisfies $h(s_{goal}) = 0$ and $h(s) \leq h(s') + c(s, s')$, $\forall s, s'$ such that $s' \in \text{SUCC}(s)$ and $s \neq s_{goal}$. OPEN denotes a priority queue ordered by some priority function such as $g(s) + h(s)$ or $g(s) + w \cdot h(s)$ with $w \geq 1$.

Definition 1 (Uncalibrated Heuristic). *An uncalibrated heuristic $h^u : S \rightarrow \mathbb{R}$ is a heuristic that induces a ranking for a set of states, i.e. state s_i is ranked higher than state s_j by the uncalibrated heuristic h^u if $h^u(s_i) > h^u(s_j)$. Note that the uncalibrated heuristic has no relation to the cost-to-go for a state and does not have non-negativity constraints.²*

MHA*

MHA* assumes that there is one consistent heuristic h_0 and n possibly inadmissible heuristics $h_i, i = 1, 2, \dots, n$. It maintains an *anchor* search (OPEN₀), where states are sorted by $f_0(s) = g(s) + w_1 \cdot h_0(s)$ and n inadmissible searches (OPEN _{i} , $i = 1, 2, \dots, n$), where states are sorted by $f_i = g(s) + w_1 \cdot h_i(s)$. Note that the g -values for all states

²More generally, the uncalibrated heuristic could be a mapping from $X \in 2^S$, the power set of states, to some $s \in X$.

are shared across the searches, allowing the algorithm to automatically figure out which heuristics are useful at different times of the search. MHA* cycles through each of the inadmissible searches and expands the state at the top of OPEN _{i} if the condition $\min_{s \in \text{OPEN}_i} f_i(s) \leq w_2 \cdot \min_{s \in \text{OPEN}_0} f_0(s)$ is met, otherwise making an expansion from the anchor search. While this condition enables MHA* to provide a sub-optimality bound of $w_1 \cdot w_2$, it also creates a calibration problem: since $h_i, i = 1, 2, \dots, n$ can be arbitrarily inadmissible, $f_i(s)$ could be completely out-of-scale (for example, several orders of magnitude larger) with $f_0(s)$, thereby never meeting the required condition for expansion. Consequently, MHA* would only expand states from the anchor search, essentially reducing to running weighted A* search with a single consistent heuristic.

Improved MHA*

Overview

Improved MHA* is designed to (a) guarantee that a solution found is within a desired sub-optimality bound, (b) make efficient use of the inadmissible heuristics, and (c) solve the calibration problem. We first provide high level intuition for each of the above three before going into the details.

An admissible heuristic provides a lower bound on the optimal solution cost $g^*(s_{goal})$, or the w -optimal solution cost $w \cdot g^*(s_{goal})$ if using an inflated admissible heuristic such as in wA* (Pohl 1970). Then given a desired sub-optimality bound, one could run any search underneath (such as a greedy search on an inadmissible heuristic) to find a feasible solution and check if the cost of that solution is less than the lower bound derived from the admissible heuristic. Since Improved MHA*, like the original MHA* has access to one consistent (and hence admissible) heuristic, it can use the procedure just described to run inadmissible greedy searches underneath, while using the admissible heuristic to provide guarantees on solution quality. However, the issue with running inadmissible or greedy searches underneath is that they could be spending all their time expanding states from where one can surely not obtain a w -optimal solution, thereby making inefficient use of the inadmissible heuristics. Improved MHA* addresses this by allowing the inadmissible searches to only expand ‘promising’ states that have some possibility of leading to a w -optimal solution. Different criteria for selecting these promising states lead to different variants of the algorithm. Finally, unlike MHA*, Improved MHA* operates greedily on the uncalibrated inadmissible heuristics, never additively combining them with the cost-to-come. This ensures that no heuristic is ‘skipped’ over, as might happen in MHA* because of the calibration problem.

Algorithm

For most parts of the algorithm, Improved MHA* resembles weighted A* (Pohl 1970) without re-expansions (Likhachev, Gordon, and Thrun 2004). In fact, if we remove lines 18-21 in Alg. 1, then the algorithm is identical to weighted A* (wA*). The difference arises in the fact that we interleave wA* expansions with ‘inadmissible’ expansions by

Algorithm 1 Improved MHA*: Requires instantiations of TERM-CRITERION(s), PRIORITY(s), P-CRITERION(s)

Inputs:

The start state s_{start} and the goal state s_{goal}
 Sub-optimality bound factor $w (\geq 1)$
 One consistent heuristic h and n arbitrary (possibly inadmissible, uncalibrated) heuristics h_1, h_2, \dots, h_n .

Output:

A path from s_{start} to s_{goal} whose cost is within $w \cdot g^*(s_{goal})$.

```

1: procedure EXPANDSTATE( $s$ )
2:   Remove  $s$  from OPEN
3:   for all  $s' \in \text{SUCC}(s)$  do
4:     if  $s'$  was not seen before then
5:        $g(s') \leftarrow \infty$ 
6:       if  $g(s') > g(s) + c(s, s')$  then
7:          $g(s') \leftarrow g(s) + c(s, s')$ 
8:         if  $s \notin \text{CLOSED}_a$  then
9:           Insert/Update  $s'$  in OPEN with PRIORITY( $s'$ )
10: procedure MAIN()
11:   OPEN  $\leftarrow \emptyset$ 
12:   CLOSEDa  $\leftarrow \emptyset$ , CLOSEDu  $\leftarrow \emptyset$ 
13:    $g(s_{start}) \leftarrow 0$ ,  $g(s_{goal}) \leftarrow \infty$ 
14:   Insert  $s_{start}$  in OPEN with PRIORITY( $s_{start}$ )
15:   while not TERM-CRITERION( $s_{goal}$ ) do
16:     if OPEN.EMPTY() then return null
17:     P-SET  $\leftarrow \{s : s \in \text{OPEN}$ 
18:        $\wedge s \notin \text{CLOSED}_u$ 
19:        $\wedge \text{P-CRITERION}(s)\}$ 
20:     for  $i = 1, \dots, n$  do
21:        $s_i \leftarrow \arg \min_{s \in \text{P-SET}} \text{RANK}(s, i)$ 
22:       EXPANDSTATE( $s_i$ )
23:       CLOSEDu  $\leftarrow \text{CLOSED}_u \cup \{s_i\}$ 
24:        $s_a \leftarrow \text{OPEN.TOP}()$ 
25:       EXPANDSTATE( $s_a$ )
26:       CLOSEDa  $\leftarrow \text{CLOSED}_a \cup \{s_a\}$ 
27:   return solution path

```

each of the n inadmissible heuristics. As in wA*, we maintain an OPEN list sorted by some priority (such as $f(s) = g(s) + w \cdot h(s)$). Then the algorithm repeatedly performs the following until a termination condition is satisfied: a subset of the OPEN list called the potential set (abbreviated as P-SET) is constructed (line 17). As explained earlier, states in the P-SET are likely to lead towards a bounded sub-optimal solution. Then, each available heuristic h_i selects a state for expansion from the P-SET according to a ranking function $\text{RANK}(s, i)$ (line 19). For an uncalibrated heuristic, this is simply $h_i(s)$, whereas for a calibrated heuristic, the ranking function is $g(s) + w \cdot h_i(s)$. These states are expanded and marked as expanded ‘inadmissibly’ (line 21). Finally, the state at the top of OPEN (s_a) is expanded ‘admissibly’ and marked as closed for the *anchor search* (lines 22-24), following the same terminology as in (Aine et al. 2014). The rationale behind making an admissible expansion during every execution of the while loop (line 15) is that we might obtain a tighter (larger) lower bound on the w -optimal solution cost, which in turn provides more freedom for the inadmiss-

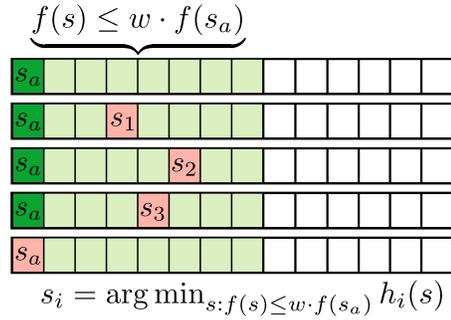


Figure 1: Illustration showing the operation of Focal-MHA*. Each row depicts the OPEN list and the states selected for expansion during an iteration over the available heuristics.

sible heuristics to ‘explore’ the state space.

As noted in the algorithm pseudocode, we need to provide instantiations for the OPEN list priority, the termination criterion, and the criterion for membership in the P-SET. We present three variants of Improved MHA* based on different instantiations of the said methods.

MHA*++ MHA*++ uses the following instantiations:

PRIORITY(s) : $g(s) + w \cdot h(s)$

TERM-CRITERION(s) : $g(s) \leq \max_{s \in \text{CLOSED}_a} \text{PRIORITY}(s)$

P-CRITERION(s) : $g(s) + h(s) \leq \max_{s \in \text{CLOSED}_a} \text{PRIORITY}(s)$

MHA*++ uses a weighted A* search as its anchor and evaluates P-SET membership by comparing the unweighted priority ($g(s) + h(s)$) of a state with the maximum weighted priority ($g(s) + w \cdot h(s)$) of any state expanded admissibly. Although typically one uses the minimum priority from OPEN to obtain bounds, we use the maximum priority from CLOSED since the anchor is a wA* search and the priorities (f -values) need not be monotonically non-decreasing as in regular A*. This way, we can maintain a monotonically non-decreasing lower bound on the w -optimal solution cost and provide maximum latitude for inadmissible expansions.

Focal-MHA* Focal-MHA* uses the following instantiations:

PRIORITY(s) : $g(s) + h(s)$

TERM-CRITERION(s) : $g(s) \leq w \cdot \min_{s \in \text{OPEN}} \text{PRIORITY}(s)$

P-CRITERION(s) : $g(s) + h(s) \leq w \cdot \min_{s \in \text{OPEN}} \text{PRIORITY}(s)$

Focal-MHA* is so named because of its direct connection with the A_ϵ^* family of algorithms and their use of the FOCAL list (Pearl and Kim 1982). Here, the anchor search is an optimal A* search and the P-SET is simply the FOCAL list, i.e., it is a prefix of the OPEN list and contains states whose f -values are within w of the best f -value in OPEN. Figure 1 shows the operation of Focal-MHA*. During every execution of the while loop (line 15 in Alg. 1), every heuristic selects and expands a state from the FOCAL list according to the RANK function, and finally the anchor state (s_a) itself is expanded.

Unconstrained-MHA* As suggested by the name, Unconstrained-MHA* imposes no restrictions for membership in the P-SET:

$$\text{PRIORITY}(s) : g(s) + w \cdot h(s)$$

$$\text{TERM-CRITERION}(s) : g(s) \leq \max_{s \in \text{CLOSED}_a} \text{PRIORITY}(s)$$

$$\text{P-CRITERION}(s) : \text{true}$$

This algorithm is similar to the multi-heuristic greedy best-first search proposed in (Röger and Helmert 2010) in that it uses each inadmissible heuristic to run an unconstrained greedy search. However, the use of the anchor search in our case enables us to guarantee bounds on solution quality.

Theoretical Analysis

All variants of Improved MHA* have guarantees similar to MHA*: the sub-optimality of the solution found is bounded by w times the cost of the optimal solution, and no state is expanded more than twice (at most once by the anchor search and at most once across all inadmissible searches). In addition, MHA*++ and Focal-MHA* provide the guarantee that if the search currently does not have a w -optimal solution through a particular state s in OPEN (i.e., the state is not ‘promising’), then s will not be expanded inadmissibly. This property is novel to Improved MHA* and distinguishes it from MHA*. These properties are formalized below:

Theorem 1. *At any point during the execution of Improved MHA* (for all its variants), $\text{PRIORITY}(s_a) \leq w \cdot g^*(s_{goal})$, where $s_a = \arg \min_{s \in \text{OPEN}} \text{PRIORITY}(s)$*

Proof. (Sketch) For MHA*++ and Unconstrained-MHA*, the proof for this theorem follows in a manner similar to the proof for wA* without re-expansions (Likhachev, Gordon, and Thrun 2004). What makes it different from wA* without re-expansions is that states can be expanded ‘out-of-order’ by the inadmissible heuristics, possibly violating the invariants maintained by wA*. However, by allowing any state to be re-expanded a second time by the anchor search, we can show that the anchor search can rectify the g -value of any state s if $g(s) > w \cdot g^*(s)$. This essentially proves that the invariant maintained by wA* without re-expansions still holds for Improved IMHA*, i.e, the priority of state s_a at the top of OPEN is a lower bound on $w \cdot g^*(s_a)$ and $w \cdot g^*(s_{goal})$. A rigorous proof for this theorem would be identical to the proofs provided for Shared MHA* (Aine et al. 2014).

For Focal-MHA*, the proof is identical to the above except that we have a stronger bound: $\text{PRIORITY}(s_a) \leq g^*(s_{goal})$. This follows from the fact that the anchor search is an optimal A* search. However, because $w \geq 1$, the theorem is trivially true for Focal-MHA* too. ■

Corollary 1. *At any point during the execution of Improved MHA* (for all its variants), $\text{PRIORITY}(s_a) \leq w \cdot g^*(s_{goal})$, where $s_a = \arg \max_{s \in \text{CLOSED}_a} \text{PRIORITY}(s)$*

Proof. CLOSED_a contains states that have been expanded by the anchor search, i.e, those states that were at the top of OPEN at some point during the search. From Theorem 1, we know that every state in CLOSED_a has a

priority that lower bounds the w -optimal solution cost. Specifically, the maximum priority of any of those states $\max_{s \in \text{CLOSED}_a} \text{PRIORITY}(s)$ is also a lower bound on $w \cdot g^*(s_{goal})$. ■

Theorem 2 (Bounded re-expansions). *No state is expanded (opened) more than twice by any variant of Improved MHA*, i.e, a state can be re-expanded (re-opened) only once.*

Proof. For any state s to be expanded from OPEN, it must first be inserted into OPEN and this happens only when either $s \notin \text{CLOSED}_u$ or $s \notin \text{CLOSED}_a$ (lines 8 and 17). Further, every expanded state s is added to either CLOSED_u or CLOSED_a (lines 21 and 24). Thus, it immediately follows that a state can be expanded at most twice before it is added to both CLOSED_u and CLOSED_a . In fact, if a state is added first to CLOSED_a before it is added to CLOSED_u , it will not be expanded a second time at all (line 8). ■

Theorem 3 (Bounded sub-optimality). *All variants of Improved MHA* terminate and when they do, the solution returned (if one exists) has a cost which is at most w times the cost of the optimal solution. In other words, when Improved MHA* terminates, $g(s_{goal}) \leq w \cdot g^*(s_{goal})$.*

Proof. The search terminates either on line 16 or line 25. Termination on line 25 occurs only when TERM-CRITERION is satisfied. Using Theorem 1 and Corollary 1 with the termination criterion for each variant, we see that the search terminates only when $g(s_{goal}) \leq w \cdot g^*(s_{goal})$, thus proving the theorem.

For the case when no solution exists, OPEN will be empty (line 16) once every state in the graph has been expanded at most twice (Theorem 2) and the search terminates. ■

Theorem 4 (Efficiency). *For MHA*++ and Focal-MHA*, any state s with $g(s) + h(s) > w \cdot g^*(s_{goal})$ will not be expanded inadmissibly.*

Proof. For MHA*++, P-SET membership requires $g(s) + h(s) \leq \max_{s \in \text{CLOSED}_a} \text{PRIORITY}(s)$. From Corollary 1, we see that states in the P-SET satisfy $g(s) + h(s) \leq w \cdot g^*(s_{goal})$. For Focal-MHA*, the anchor search is an optimal A* search and thus $\min_{s \in \text{OPEN}} \text{PRIORITY}(s) \leq g^*(s_{goal})$. Using this in the P-SET membership criterion, we see that all states in the P-SET satisfy $g(s) + h(s) \leq w \cdot g^*(s_{goal})$. Thus, in MHA*++ as well as Focal-MHA*, a state with $g(s) + h(s) > w \cdot g^*(s_{goal})$ cannot belong to the P-SET, and can thus never be expanded inadmissibly. ■

Experimental Results

Motion Planning

We first evaluate the performance of the Improved MHA* variants on an 11 degree of freedom (DoF) full-body robot motion planning domain for the PR2 (a dual-arm mobile robot). The objective is for the planner to generate a collision free motion for the robot to approach and pick up objects on cluttered tables in a kitchen environment. In addition to controlling the right arm of the robot (in order to grasp the object) the robot generally starts out of reach and must drive

Table 1: Comparison of different Improved MHA* variants with the original MHA* algorithm, MH-GBFS and RRT-Connect for full-body motion planning.

	$w = 100$				$w = 10$				$w = 5$				MH-GBFS	RRT-C
	++	Focal	Uncons	Orig.	++	Focal	Uncons	Orig.	++	Focal	Uncons	Orig.		
Success (%)	84	75	84	61	83	74	83	0	66	74	60	0	85	45
States Expanded	2415	3058	2415	5179	3293	3086	3227	-	2378	3086	2472	-	2752	n/a
Plan Time (s)	36.89	47.44	36.98	75.63	47.60	47.88	46.96	-	37.92	48.08	38.31	-	41.98	21.95
Base Cost (m)	5.33	5.47	5.33	5.47	5.32	5.52	5.33	-	4.77	5.52	4.49	-	5.45	5.22
Arm Cost (rad)	6.32	6.45	6.32	6.02	6.17	6.49	6.17	-	5.70	6.49	5.41	-	6.60	8.26

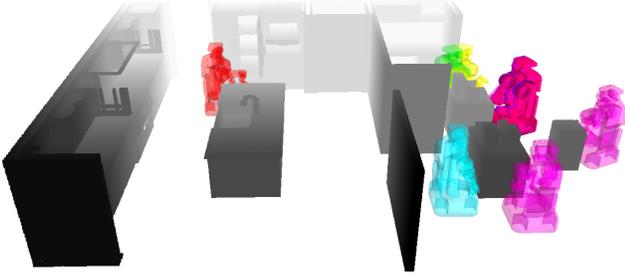


Figure 2: The kitchen domain for our experiments. Note the randomly placed tables on the right with randomized clutter on top. In the middle is a narrow door that separates the two rooms. The different colored robots show node expansions from each of the 20 different heuristics we used.

to the table before reaching. Specifically, the planner controls the base’s position and orientation (x, y, θ) , the height of the prismatic spine which raises and lowers the torso, the 6 DoF pose of the gripper in the robot’s body frame $(x_{hand}, y_{hand}, z_{hand}, roll_{hand}, pitch_{hand}, yaw_{hand})$, and the arm’s “free angle” (which way the elbow is pointing).

Each state in the graph we plan on corresponds to a complete robot configuration. From any state the robot has a set of *motion primitives* it can apply, which are short kinematically feasible motions (Likhachev and Ferguson 2008; Cohen, Chitta, and Likhachev 2010). Collision free motion primitives connect pairs of states, thereby representing edges in our graph. While there is a single start state, the goal state is underspecified as any state that results in the gripper reaching the object meets the goal conditions—for instance, the robot could pick up the object from different (x, y, θ) locations around the object.

Figure 2 shows the environment we ran our experiments in. The domain is challenging due to the high dimensionality of the problem, cluttered tables, and narrow passages which must be crossed (the robot’s base barely fits through the doorway and only if the arm is tucked in). Determining a single heuristic which can guide the base and arm toward the goal and around obstacles is challenging. It becomes especially hard when there are conflicting ideas, like wanting to extend the arm when reaching for the goal, but wanting to tuck it when going through a door. A multi-heuristic search is perfect for dealing with these heuristics with multiple (and at times conflicting) components.

We designed 20 heuristics (19+1 anchor) to help guide the search. 16 of these guide the base’s (x, y) position while requiring different fixed base headings and a tucked arm. These heuristics help navigation in tight spaces, but can’t reach for the goal. There are then 3 other heuristics which guide the arm to the goal with or without guiding the base to a specific pose within arm’s reach of the goal. One guides the base to a pose “behind” the goal (so that the gripper faces forward when the robot gets there), the second focuses on gripper orientation, while the third only tries to pull the gripper to the proper position and orientation without influencing the base position. Note that almost all of these heuristics are inadmissible and uncalibrated. They serve as ranking functions rather than estimates of cost-to-go—e.g., the tuck-arm heuristic merely prefers to expand states where the robot’s arm is tucked in as opposed to other states.

We generated 100 random trials. The two tables in the kitchen are randomly positioned differently every 10 trials as is the clutter on top of them. Each of the 100 trials is created by choosing a random pose on one of the two tables for the gripper to reach and the starting configuration for the robot is randomly generated as well. A trial is deemed successful if the planner can find a w -optimal solution within a time limit of 5 minutes, and unsuccessful otherwise.

Table 1 compares the three variants, MHA*++, Focal-MHA* and Unconstrained-MHA* with the original MHA* algorithm for different sub-optimality bounds w , as well as the multi-heuristic greedy best-first search (MH-GBFS) (Röger and Helmert 2010) and RRT-Connect (Kuffner and LaValle 2000). To uniformly compare across all methods, the solution quality of the generated paths is measured by the distance traveled by the robot base and the arm (joint angles). The reported statistics for a method are average values across its successful trials.

Unlike the Improved MHA* variants, the original MHA* algorithm requires two sub-optimality factors w_1 and w_2 , for the inflation and anchor respectively. As recommended in (Aine et al. 2014), we set $w_2 = \min(2.0, \sqrt{w})$ and $w_1 = w/w_2$ for all our comparisons to get the same desired sub-optimality bounds for each case. The Improved MHA* methods significantly outperform the original MHA* algorithm for lower sub-optimality bounds; in fact the original MHA* algorithm fails to succeed on any trial at all. This is expected, since MHA* essentially reduces to weighted A* with a single heuristic when the inadmissible heuristics are out-of-scale (several orders of magnitude greater)

with the consistent heuristic, or equivalently when the anchor sub-optimality factor (w_2) is too small. For a large sub-optimality bound ($w = 100$) however, MHA* provides a reasonable success rate as was shown in (Aine et al. 2014).

MH-GBFS performs comparably to Unconstrained-MHA* for larger sub-optimality bounds as expected, since they both run unconstrained greedy searches. The high success rate of these approaches can be attributed to the fact that the inadmissible heuristics designed for this problem are all useful at some point or another, thereby not really requiring the ‘control’ provided by MHA*++ and Focal-MHA* when operating at higher sub-optimality bounds. However, when we desire lower sub-optimality bounds, it becomes essential to control the inadmissible searches, as can be seen from Table 1 for $w = 5$. Characteristic of greedy search, MH-GBFS has higher solution costs especially when compared to MHA*++ and Unconstrained-MHA*.

RRT-Connect (Kuffner and LaValle 2000) is a popular sampling-based motion planning algorithm in robotics. While it is known to quickly generate plans for high dimensional problems, it suffers from a ‘narrow passage’ problem. In our experiments, the doorway in the kitchen creates a narrow passage in the 11-DoF configuration space, thereby affecting RRT-Connect’s success rate. Moreover, RRT-Connect does not explicitly minimize a cost and therefore the paths generated typically have high cost (Table 1).

Sliding Tile Puzzles

In this section, we present the experimental results for large sliding tile puzzles (8×8 , 9×9 and 10×10). For each size, we create 100 random (solvable) puzzle instances to build our test suite. We evaluate the performance of the Improved MHA* variants and MH-GBFS over the entire test suite. In each case, we run the planner for a time limit of 5 minutes.

For this domain, we used a set of 9 heuristics (8 inadmissible + 1 anchor). We used the Manhattan distance plus linear conflicts as the consistent heuristic (anchor). The inadmissible heuristics were computed in the following manner: for a given puzzle size, we generate a database of 1000 different solved configurations by performing a random walk of k steps from the goal state, where k is a random number between 2 and 10 times the puzzle size. For each configuration, we store the path to goal and store k as the cost to goal.

We cluster this database in 8 parts using the heuristic difference between two configurations as the distance metric. For a given instance to solve (say with configuration s_c), we pick one target configuration (tc_i) from each cluster, such that the heuristic distance between s_c and tc_i is minimum. Once a target configuration tc_i is chosen, inadmissible heuristic h_i for any state s was computed by $h_i(s) = w \cdot h_0(s, tc_i) + cost(tc_i)$ (note that this heuristic includes inflation), where w is the desired sub-optimality bound. For the original MHA* algorithm we set $w_2 = \min(2.0, \sqrt{w})$ and $w_1 = w/w_2$ (we use $h_i(s) = w_1 \cdot h_0(s, tc_i) + cost(tc_i)$). It may be noted that unlike the full-body planning domain, the inadmissible heuristics for this domain are not really uncalibrated as these are computed using the same function as the consistent heuristic (albeit with different target configurations). Therefore in this case, we use $g + h$ ranking (we do

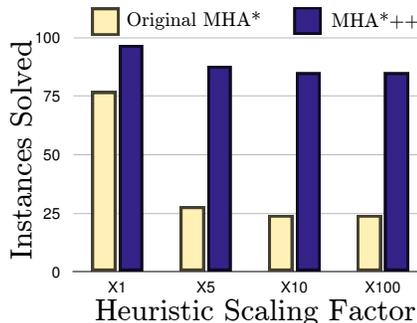


Figure 3: Comparison between the original MHA* and MHA*++ for 9×9 puzzles ($w = 10$) for different scalings of the heuristic. The x -axis shows the factors with which the inadmissible heuristics are multiplied (e.g., $\times 5$ denotes multiplication by 5) and the y -axis shows the number of instances solved (out of 100), within a time limit of 5 minutes.

not inflate the heuristics here, as they are already inflated) for the Improved MHA* variants (line 19 in Alg. 1), as it takes into account the impact of g values.

We include the results for this domain in Table 2. The first thing to note is that original MHA* does not perform as poorly as in the full-body planning domain. This is expected, as the heuristics used in this domain are not really out-of-scale. However, even in this case, MHA*++ consistently performs better/equivalent to the original on most trials, and the improvement gets more pronounced with larger puzzle size and lower desired sub-optimality bounds. As examples, for the 9×9 puzzle with $w = 5$, original MHA* solves 61 instances whereas MHA*++ solves 87, and for 10×10 with $w = 5$, original MHA* solves 21 instances whereas MHA*++ solves 42. This highlights the fact that even if we have heuristics that are not out-of-scale, MHA*++ can dominate original MHA* due to its improved control, ranking, and expansion policies. Considering the other variants, we observe that MHA*++ and original MHA* tend to do better than others in most cases indicating that when the heuristic is not out-of-scale, weighted best-first ranking is probably a better choice than greedy ranking. However, there are cases where the opposite is true (as seen for 9×9 with $w = 10$).

To understand the impact of out-of-scale heuristics in the puzzle domain, we did the following experiment: we multiplied the inadmissible heuristics by a chosen factor (5, 10 and 100) and ran original MHA* and MHA*++ for the 9×9 puzzle with $w = 10$. The results (Fig. 3) clearly depict the impact of out-of-scale heuristics on original MHA*; its performance degrades considerably as we make the heuristics more out-of-scale, and after a point (10 and above) it reduces to weighted A* (with additional overhead of multiple queue updates). In contrast, MHA*++ remains robust to heuristic scaling and outperforms MHA* by a significant margin.

Discussion

The experimental results show that different Improved MHA* variants are useful under different circumstances.

Table 2: Comparison of different Improved MHA* variants with the original MHA* algorithm and MH-GBFS for sliding tile puzzle problems.

Size		$w = 50$				$w = 10$				$w = 5$				MH-GBFS
		++	Focal	Uncons	Orig.	++	Focal	Uncons	Orig.	++	Focal	Uncons	Orig.	
8×8	Success (%)	95	91	91	97	100	92	96	100	100	66	61	83	94
	Plan Time (s)	18.16	38.56	27.74	20.70	18.23	28.03	22.65	18.14	19.56	39.43	42.95	26.31	28.45
	Sol. Cost	1617.4	1883.6	1704.0	1573.1	1552.5	1848.9	1656.4	1438.4	1415.6	1642.5	1345.2	1302.7	1889.8
9×9	Success (%)	94	69	85	93	97	61	86	77	87	46	28	61	71
	Plan Time (s)	44.25	53.52	60.49	43.15	46.99	60.31	53.16	62.16	58.78	49.18	66.26	89.33	55.59
	Sol. Cost	2127.6	2374.8	2350.6	2030.8	1988.6	2428.2	2196.8	1914.0	1750.2	2084.6	1560.2	1630.7	2447.0
10×10	Success (%)	41	24	35	44	51	27	34	36	42	19	10	21	24
	Plan Time (s)	89.84	81.62	88.50	88.33	93.62	90.00	92.78	98.42	100.12	88.28	91.00	114.40	99.22
	Sol. Cost	2523.1	2951.8	2800.3	2411.4	2449.9	2859.7	2571.5	2261.0	2251.9	2533.3	2075.4	1994.2	2663.4

The two influential factors in deciding which variant to use are (i) the desired sub-optimality bound for the solution and (ii) the quality of the one consistent heuristic, i.e, how close h is to the perfect heuristic. Focal-MHA* uses an optimal A* search for its anchor whereas MHA*++ and Unconstrained-MHA* use a weighted A* search for their anchor. For Focal-MHA*, the size of P-SET is determined only by the state at the top of OPEN and the desired sub-optimality bound (w), unlike the other two where the quality of the heuristic also affects the size of the P-SET. Thus, one can expect Focal-MHA*'s behavior to be more consistent across different sub-optimality bounds as opposed to the other two, since there is no interaction between w and h .

While it might seem that Unconstrained-MHA* does as well as MHA*++ from Table 1, there can be cases where MHA*++ is clearly better. This is true especially when we desire low sub-optimality bounds or when the inadmissible heuristic used is uninformative. Figure 4 shows a synthetic example of a 2D Manhattan world planning problem where MHA*++ performs much better than Unconstrained-MHA* in the presence of a spurious inadmissible heuristic.

Finally, unlike MHA* which uses two sub-optimality factors w_1 and w_2 , Improved MHA* requires only a single factor w . This permits a natural extension to an anytime algorithm that can improve its solution quality over time. Furthermore, we would like to note that alternative scheduling strategies (Phillips et al. 2015) could be used to cycle through the set of available heuristics, instead of the round-robin scheme presented here (line 18 in Alg. 1).

Conclusions

We presented a general multi-heuristic search framework, Improved MHA*, which can use multiple inadmissible and uncalibrated heuristics in a synergetic fashion to find a solution. We presented three variants of the framework and for each, provided theoretical guarantees on completeness, bounded sub-optimality and bounded re-expansions. Additionally, we proved that two of the variants can use the inadmissible heuristics efficiently by focusing them only on promising states. We evaluated our algorithm on complex 11-DoF robotics motion planning problems and sliding tile puzzles and showed that it significantly outperforms the

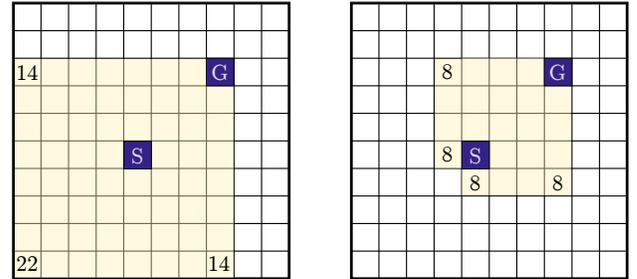


Figure 4: An example showing the advantage of MHA*++ over Unconstrained-MHA* in the presence of a spurious inadmissible heuristic. Assume that for this 2D Manhattan world problem, the consistent heuristic h is the perfect heuristic and that there is one inadmissible heuristic h_1 which prefers expanding states closer to the lower left corner. Let the desired sub-optimality bound $w = 1.5$. For this example, $g^*(s_{goal}) = 6$ and $w \cdot g^*(s_{goal}) = 9$. The shaded cells show the states which can be expanded inadmissibly by h_1 for Unconstrained-MHA* (left) and MHA*++ (right). Also marked on select cells is the $g^*(s) + h^*(s)$ value, the cost of the optimal path through that cell. As seen from the illustration, Unconstrained-MHA* can spend a lot of effort expanding states which cannot lead to a w -optimal solution.

original MHA* algorithm.

The use of multiple heuristics for search naturally raises the possibility of parallelization. While A* is a fundamentally sequential algorithm, we believe that the structure provided by Improved MHA* can be exploited for parallel expansion of states from the P-SET. Finally, we anticipate that Improved MHA* will provide a platform for the design and use of uncalibrated heuristics that have no relation with the cost-to-go, but might effectively guide best-first searches.

Acknowledgments

We would like to thank the anonymous reviewers for their helpful suggestions and feedback. This research was sponsored by the ONR DR-IRIS MURI grant N00014-09-1-1052.

References

- Aine, S.; Swaminathan, S.; Narayanan, V.; Hwang, V.; and Likhachev, M. 2014. Multi-heuristic A*. In *Proceedings of Robotics: Science and Systems*.
- Betz, C., and Helmert, M. 2009. Planning with h+ in theory and practice. In *KI 2009: Advances in Artificial Intelligence*. Springer. 9–16.
- Cohen, B. J.; Chitta, S.; and Likhachev, M. 2010. Search-based planning for manipulation with motion primitives. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, 2902–2908. IEEE.
- Ebendt, R., and Drechsler, R. 2009. Weighted A* search—unifying view and application. *Artificial Intelligence* 173(14):1310–1342.
- Hart, P. E.; Nilsson, N. J.; and Raphael, B. 1968. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics* 4(2):100–107.
- Hatem, M., and Ruml, W. 2014. Simpler bounded suboptimal search. In *Proceedings of Twenty-Eighth AAAI Conference on Artificial Intelligence*. AAAI Press.
- Hoffmann, J. 2005. Where ‘ignoring delete lists’ works: Local search topology in planning benchmarks. *Journal of Artificial Intelligence Research* 685–758.
- Isto, P. 1996. Path planning by multiheuristic search via subgoals. In *Proceedings of the 27th International Symposium on Industrial Robots, CEU*, 71272–6.
- Kuffner, J. J., and LaValle, S. M. 2000. Rrt-connect: An efficient approach to single-query path planning. In *ICRA*, 995–1001. IEEE.
- Likhachev, M., and Ferguson, D. 2008. Planning long dynamically-feasible maneuvers for autonomous vehicles. In *Proceedings of Robotics: Science and Systems (RSS)*.
- Likhachev, M.; Gordon, G. J.; and Thrun, S. 2004. ARA*: Anytime A* with provable bounds on sub-optimality. In *Advances in Neural Information Processing Systems 16*. Cambridge, MA: MIT Press.
- Pearl, J., and Kim, J. H. 1982. Studies in semi-admissible heuristics. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* (4):392–399.
- Phillips, M.; Narayanan, V.; Aine, S.; and Likhachev, M. 2015. Efficient search with an ensemble of heuristics. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*. AAAI Press (to appear).
- Pohl, I. 1970. First results on the effect of error in heuristic search. *Machine Intelligence* 5:219–236.
- Röger, G., and Helmert, M. 2010. The more, the merrier: Combining heuristic estimators for satisficing planning. In *ICAPS’10*, 246–249.
- Thayer, J. T., and Ruml, W. 2011. Bounded suboptimal search: A direct approach using inadmissible estimates. In *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, 674–679.