On the Properties of All-Pair Heuristics

Shahaf S. Shperberg[1], **Ariel Felner**[1], **Lior Siag**[1], **Nathan R. Sturtevant**[2,3]
[1] Ben-Gurion University of the Negev
[2] University of Alberta
[3]Alberta Machine Intelligence Institute (Amii)
{shperbsh, felner}@bgu.ac.il, siagl@post.bgu.ac.il, nathanst@ualberta.ca

Abstract

While most work in heuristic search concentrates on *goal-specific* heuristics, which estimate the shortest path cost from any state to the goal, we explore *all-pair* heuristics that estimate distances between all pairs of states. We examine the relationship between these heuristic functions and the shortest distance function they estimate, revealing that all-pair consistent heuristics may violate the triangle inequality. Thus, we introduce a new property for heuristics called Δ-consistency, requiring adherence to the triangle inequality. Additionally, we present a method for transforming standard consistent heuristics to be Δ-consistent, showcasing its benefits through a synthetic example. We then show that common heuristic families inherently exhibit Δ-consistency. This positive finding encourages the use of all-pair consistent heuristics, and prompts further investigation into the optimality of A*, when given an *all-pair* heuristic instead of a *goal-specific* heuristic.

Introduction

The aim of A* (Hart, Nilsson, and Raphael 1968) and its many derivatives is to find a least-cost path between a given start state (vertex) and a goal state (or set of vertices) in a graph representing combinatorial and pathfinding problems. A* explores states in a best-first search order according to $f(n) = g(n) + h(n)$ where $g(n)$ is the currently known cheapest cost from the start state to n and $h(n)$ is a heuristic function that estimates the cost-to-go from n to the goal. A* is guaranteed to return an optimal solution if the heuristic is *admissible*. A desirable attribute of an admissible heuristic is *consistency*, which ensures that A* never explores the same state more than once.

Most of the literature on consistent heuristics, including the optimally-efficient proof of A* (Dechter and Pearl 1985), has focused on a heuristic computed specifically toward the goal (denoted here as a *goal-specific heuristic*). In this paper, we delve deeper into the concept of *consistency*, assuming that an admissible consistent heuristic is defined for *all pairs of states* in the underlying graph, not solely directed towards the goal. An *all-pair consistent heuristic* $h(u, v)$ estimates the shortest distance $d(u, v)$.

Examination of all-pair consistent heuristics reveals an intriguing observation: some consistent heuristics may not conform to the triangle inequality. To explore this phenomenon, we adopt a broader perspective and compare the similarities and discrepancies of the d and h functions across various properties in relation to metric spaces. We discuss existing methods for strengthening h and d, rectifying some of the discrepancies between them. Furthermore, we distinguish between a *consistent heuristic*, which can be seen as a variation of the triangle inequality defined with respect to both h and d, and a stronger attribute which we denote as Δ-*consistent heuristic*, where the heuristic function adheres to the triangle inequality defined solely with respect to h. We introduce a novel approach, denoted as Heuristic-Differential Heuristic (HDH), that enforces a consistent heuristic to also be Δ-consistent by increasing the h-values of some pairs of states. We demonstrate how HDH reduces the search effort in a synthetic example. Interestingly, this example serves to highlight that A*, when provided with an all-pair heuristic and treated as if it were a goal-specific heuristic—a practice commonly adopted—is not optimally efficient, compared to algorithms capable of leveraging pairwise heuristic estimations.

Finally, we analyze common existing heuristic families and prove they are Δ-consistent. As a result, our new HDH does not enhance any of the analyzed heuristics. In fact, we have not yet identified any state-of-the-art heuristic techniques that are consistent but not Δ-consistent. Nevertheless, this paper opens a new research direction concerning the usage of all-pair consistent heuristics and the theoretical attributes of related algorithms.

Definitions and Background

Let $I = (G = (V, E), c, s, g, h)$ be a search problem instance, where G is a graph in which V is the set of vertices, E is the set of edges, and $c : E \rightarrow \mathbb{R}^{\geq 0}$ is a cost function that assigns a non-negative cost to each edge in E. In addition, s and g are two graph vertices. The objective in heuristic search is to find a path of minimal cost from the start state s to the goal state g. Let $d : V \times V \rightarrow \mathbb{R}^{\geq 0} \cup \{\infty\}$ be the distance between vertices. That is, $d(u, v)$ represents the shortest path cost between u and v in G with regard to c (or infinity if no such path exists). The heuristic function h estimates distances between states of the graph. A heuristic is *admissible* if it is always a lower bound on the distance it estimates. We distinguish between two types of heuristics.

Goal-Specific Heuristic

A *goal-specific heuristic* estimates the distances to a specific goal node g (or set of goals) from any given state, $h_g : V \rightarrow \mathbb{R}^{\geq 0} \cup \{\infty\}$. h_g is admissible if $\forall u : h_g(u) \leq d(u, goal)$. A* and its many derivatives search towards a specific goal (or a set of goals) and use such a heuristic. Another desirable attribute for heuristics is *consistency*:

$$\forall\, u, v \in V : h_g(u) \leq d(u, v) + h_g(v) \quad (1)$$

In undirected graphs a consistent heuristic entails that $\forall(u, v) : |h(u) - h(v)| \leq d(u, v)$ (Felner et al. 2011). Given a consistent heuristic, the f-value along paths are monotonically non-decreasing. This assures that when a node u is chosen for expansion by A*, then the shortest path to u has been discovered (i.e., $g(u) = d(u)$). This rules out the need to re-open/re-expand any node. Nevertheless, admissible but inconsistent heuristics also have benefits as large heuristic values can be propagated to their neighbors through the use of *bidirectional pathmax* (Felner et al. 2011), reducing the search effort in many cases. We note that most of the literature on heuristics is on goal-specific heuristics.

All-Pair Heuristic

An *all-pair heuristic* $h : V \times V \rightarrow \mathbb{R}^{\geq 0} \cup \{\infty\}$ is defined for every pair of states u and v and is *admissible* if:

$$\forall\, u, v \in V : h(u, v) \leq d(u, v) \quad (2)$$

All-pair heuristics may be used when solving multiple instances, each time targeting a different goal state. For example, GPS navigation systems must find paths to different addresses. The *airline distance* heuristic, sometimes employed by GPS systems, can be easily applied between any two states. Yet, within a search towards a given goal, such heuristics are usually used as goal-specific heuristics.

Additionally, all-pair heuristics might be needed for some variants of Bidirectional heuristic search (BiHS). BiHS searches from both directions (from start to the goal and from the goal to start) and tries to find a meeting point of both search frontiers. A *Front-to-end* heuristic is based on two goal-specific heuristics, one for the forward search towards the goal ($h_{goal}(n)$) and one for the backward search towards the start ($h_{start}(n)$). A *front-to-front heuristic* (Kaindl and Kainz 1997; de Champeaux and Sint 1977; Eckerle et al. 2017; Siag et al. 2023) estimates the distance between any two states that can be in the two frontiers, and is equivalent to an all-pair heuristic.

An all-pair heuristic h is *consistent* if the following two equations hold:

$$\forall\, u, v, p \in V : h(u, v) \leq d(u, p) + h(p, v) \quad (3a)$$

$$\forall\, u, v, p \in V : h(u, v) \leq h(u, p) + d(p, v) \quad (3b)$$

These equations are direct interpretations of *bi-consistent heuristics* defined for bidirectional search (Eckerle et al. 2017), defined here for ordinary unidirectional search. We note that the two equations are needed because h is defined for both distances to p and distances to v and both should be considered in the equations. That is, all-pair consistency assures that the path estimation from u to v via p will never be shorter than the direct estimation of the path from u to v (as represented by the left-hand side of the equations).

Notably, heuristics found in existing literature are mostly used in unidirectional search settings and regarded as goal-specific heuristics employed within the framework of A* or its various derivatives. Nevertheless, many of these heuristics can be practically computed between all pairs. Importantly, the proof that A* is optimally efficient only considers goal-specific heuristics and does not deal with all-pair heuristics. In fact, our paper will present an example where A*, equipped with a consistent (and admissible) all-pair heuristic, unnecessarily expands nodes during the search.

Consistent and Δ-consistent Heuristics

Consistency, and in particular all-pair consistency, is often linked to the triangle inequality property. Although they share a noticeable resemblance, they are not entirely equivalent in mathematical terms. The triangle inequality applies to an individual function $f(\cdot, \cdot)$, considering three vertices u, v, p and is expressed as $f(u, v) \leq f(u, p) + f(p, v)$. It means that the direct path from u to v is no worse than the bypass from u to v via an intermediate vertex p, when measured by f. But, the consistency definition (equations 1,3) mixes two functions, d and h. The mathematically-correct triangle inequality for a heuristic h is defined as follows:

$$\forall u, v, p \in V : h(u, v) \leq h(u, p) + h(p, v) \quad (4)$$

Given an admissible and consistent heuristic h it is possible that h may not satisfy the triangle inequality. In other words, there can be three states u, v, p such that $h(u, v) > h(u, p) + h(p, v)$. Consider a minimal example of this in the graph with three vertices in Figure 2(a), where the blue lines represent the cost of the shortest paths (d-value) in G. All six shortest distances equal 20. The dashed orange lines in Figure 2(b) represent the heuristics between relevant pairs of states. All heuristics in Figure 2(b) are admissible because they are all smaller than 20. Additionally, they are all all-pair consistent as defined in equations 3a and 3b. For example, (3a) $h(u, v) = 6 \leq d(u, p) + h(p, v) = 20 + 3 = 23$, and (3b) $h(u, v) = 6 \leq h(u, p) + d(p, v) = 1 + 20 = 21$. The same reasoning applies to all other heuristic values in the figure. However, in this graph, the triangle inequality does not hold as $h(u, v) = 6 > h(u, p) + h(p, v) = 1 + 3 = 4$.

We thus distinguish between two types of all-pair consistent heuristics: standard consistent heuristics (as defined in Equations 3a and 3b), which retain the name *consistent* heuristic, and Δ-*consistent* heuristics. A Δ-*consistent* heuristic h is an admissible heuristic that also satisfies the triangle inequality with respect to V. That is, a heuristic h is Δ-*consistent* if it is admissible and satisfies Equation 4.

Δ-consistency is a stronger attribute than standard consistency, as we next prove.

Lemma 1. *If a heuristic h is Δ-consistent, it must also be consistent.*

Proof. We prove it for equation 3a. $h(u, v) \leq h(u, p) + h(p, v)$. Since h is admissible[1] then $h(u, p) \leq d(u, p)$. Thus,

[1]In general, triangle inequality does not imply admissibility.

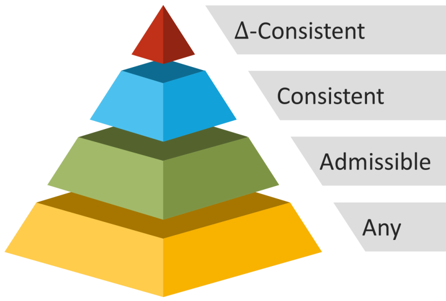


Figure 1: Illustration of the all-pair heuristic hierarchy

$h(u, v) \leq d(u, p) + h(p, v)$. The same reasoning applies for equation 3b. □

The other direction is not true — a consistent heuristic h might not necessarily be Δ-consistent, as demonstrated in our example of Figure 2.

Hence, we establish a hierarchy among all-pair heuristics, illustrated in Figure 1, ranging from weaker conditions (bottom) to stronger conditions (top). In goal-specific heuristics, there is no notion of triangle inequality (as h is not defined for all pairs of vertices), rendering Δ-consistency irrelevant in this context.

Properties of d and h

Both d and all-pair heuristics h are defined over all pairs of states. While d represents the ground truth distances, h aims to estimate d. In mathematics, various properties, such as the triangle inequality, govern distances between elements in sets. To broaden the analysis, we next explore these properties for both d and h.

Properties of d

The costs of shortest paths between pairs of vertices (represented by the function d) in directed graphs are *quasi-pseudo-metric* (Chartrand and Tian 1997). Meaning that d has the following three properties with respect to V:

1. Non-negativity: $d(u, v) \geq 0$ for all $u, v \in V$

2. Zero Self-distance: $d(u, u) = 0$ for all $u \in V$

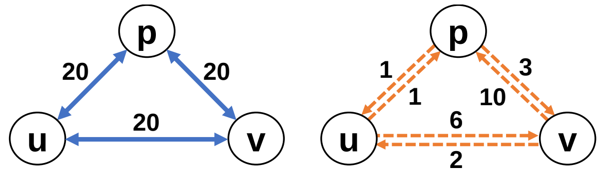
3. Triangle Inequality: $d(u, v)) \leq d(u, p) + d(p, v)$ for all $u, v, p \in V$. This means that going directly from u to v is never worse than going from u to v via a bypass through a pivot p. Obeying the triangle inequality is entailed directly from the definition of d.

Notably, if all edges are positive, d has yet another property:

4. Identity of Indiscernibles: $d(u, v) > 0$ iff $u \neq v$ for all $u, v \in V$

A function that satisfies properties 1-4 is a *quasi-metric* with respect to V. Moreover, if the graph G is undirected, then d has yet another property:

Consider a triangle where all h-values are 100 and the real distances are all smaller than 100. Clearly, the triangle inequality is valid here but the heuristics are all inadmissible. Nevertheless, our definition of Δ-consistency requires the admissibility of h.



(a) (b)

Figure 2: (a) Distances, and (b) their heuristics

5. Symmetry: $d(u, v) = d(v, u)$ for all $u, v \in V$

A function that has properties 1-3, 5 is known as *pseudo-metric*, whereas a function that possesses all five properties is known as a full *metric*.

Properties of h

Given an admissible and consistent all-pair heuristic h, we examine the properties of h w.r.t. V and d.

1. Non-negativity. By definition, $h(u, v) \geq 0$ for all $u, v \in V$.

2. Zero Self-distance. By admissibility, $h(u, v) \leq d(u, v)$. As $d(u, u) = 0$, it follows that $h(u, u) = 0$ for all $u \in V$.

3. No Triangle Inequality.

As shown above, consistency does not imply Δ-consistent. Thus, a consistent heuristic may not obey the triangle inequality.

4. No Identity of Indiscernibles. Additionally, it is possible that $h(u, v) = 0$ for some states $u \neq v \in V$, even when all edges are positive (no Property 4).

5. No Symmetry. Finally, h may not be symmetric (no Property 5), even in undirected graphs. In Figure 2(b), $h(u, v) = 2 \neq h(v, u) = 6$.

As shown, d can exhibit properties ranging from a quasi-pseudo-metric to metric, depending on the context. However, consistent heuristics do not inherently possess any of these properties. Thus, we now explore methods to adjust h to incorporate some of the properties of d. While the methods to establish Properties 4 and 5 are known, the method for establishing Property 3 is novel to this work.

Establishing Identity of Indiscernibles As mentioned above, if all edges are positive, then $d(u, v) > 0$ iff $u \neq v$. However, even when edges are positive, there may be states u, v for which $h(u, v) = 0$ and $u \neq v$. For example, the zero heuristic is both admissible and consistent. If the cost of the cheapest edge in the graph, denoted by ϵ, is known, then we can adapt the heuristic as follows $h_\epsilon(u, v) = max(\epsilon, h(u, v))$. This adaptation maintains the admissibility and consistency of h and establishes the Identity of Indiscernibles. Notably, this adaptation can be applied to goal-specific heuristics as well.

Establishing Symmetry in undirected graphs In undirected graphs, $d(u, v) = d(v, u)$. However, h may not be symmetric, i.e., states u and v may exist, such that $h(u, v) \neq h(v, u)$. Due to admissibility, we have $h(u, v) \leq d(u, v)$ and $h(v, u) \leq d(v, u)$. Consequently, if G is undirected, we can define a new heuristic $h_\leftrightarrow(u, v) = h_\leftrightarrow(v, u) =$

$\max(h(u,v),h(v,u)) \leq d(u,v) = d(v,u)$. $h_{\leftrightarrow}$ is guaranteed to be admissible and symmetric. Furthermore, if h is consistent, then $h_{\leftrightarrow}$ is also consistent, as directly derived from the definition of consistency.

While a consistent heuristic is not a quasi-pseudo-metric, as it does not guarantee the triangle inequality, a Δ-consistent heuristic qualifies as quasi-pseudo-metric w.r.t. V (or quasi-metric, if it also has the identity of indiscernibles), similar to d. Additionally, in an undirected graph where h is symmetric (e.g., using the rectification in the previous section), a Δ-consistent is pseudo-metric (or a metric, if h possesses the identity of indiscernible property).

We next propose a method, called *heuristic-differential heuristic* (HDH), for enforcing the triangle inequality in a consistent heuristic, increasing many of its h-values and making it Δ-consistent.

Strengthening Consistent Heuristics by HDH

Given a consistent any-pair heuristic h, the first inequality in Eq. 3a ($h(u,v) \leq d(u,p) + h(p,v)$) implies:

$$h(u,v) - h(p,v) \leq d(u,p) \mid \forall\, u,v,p \in V \quad (5)$$

Similarly, the second inequality in Eq. 3a ($h(u,v) \leq h(u,p) + d(p,v)$) implies:

$$h(u,v) - h(u,p) \leq d(p,v) \mid \forall\, u,v,p \in V \quad (6)$$

If, we rename the vertices in Equation 6 and set: $p = u$, $u = v$, $v = p$, we get that:

$$h(v,p) - h(v,u) \leq d(u,p) | \,\forall\, u,v,p \in V \quad (7)$$

By considering Eq. 5 and Eq. 7 we define an alternative heuristic from u to the goal p, using a pivot v, as follows:

$$\hat{h}(u,p) = \max_{v \in G}\{\max \begin{Bmatrix} h(u,v) - h(p,v) \\ h(v,p) - h(v,u) \end{Bmatrix}\} \quad (8)$$

It can be readily observed that $\hat{h}$ satisfies the triangle inequality. If we fix $v = g$ in $\hat{h}(u,v)$ then we get an alternative *goal-specific heuristic* from any state u to the goal that may be better than $h(u,g)$.

We note that calculating $\hat{h}(u,g)$ considering all possible pivot states in the graph for v (as denoted in Eq. 8) may not be practical. However, this inequality holds even when limiting v to be a subset of the states in the graph, e.g., predetermined "pivots" or only using states already discovered during the search as pivots.

In addition, in undirected graphs, we can rectify h to ensure that $h(u,v) = h(v,u)$ for all states (as described above), and utilize the fact that $d(u,v) = d(v,u)$ to get a further improved heuristic, in which terms in the maximum expression are replaced with their absolute values:

$$\hat{h}(u,p) = \max_{v \in G}\{|h(v,p) - h(v,u)|\} \quad (9)$$

This concept is reminiscent of the *differential heuristic* (DH) (Sturtevant et al. 2009), which shares a similar definition but operates on the true distances from a node v to all other nodes in the graph, rather than on the heuristics, defined as $DH_v(u,g) = |d(v,g) - d(v,u)|$. Thus, we define the Heuristic-Differential Heuristic (HDH) as follows. Let $h_{\text{HDH}}(u,v) = \max(h(u,v), \hat{h}(u,v))$. This heuristic remains admissible and consistent while also adhering to the triangle inequality, rendering it Δ-consistent.

Notably, h_{HDH} will not improve h if h is already Δ-consistent, as shown in the following lemma.

Lemma 2. *If h is Δ-consistent, then for every pair of states u and p it holds that $h_{HDH}(u,p) \leq h(u,p)$.*

Proof. For every two states u, v if $h_{\text{HDH}}(u,p) = h(u,p)$ the lemma holds trivially. Therefore, we focus on proving the lemma for the case where $h_{\text{HDH}}(u,p) = \hat{h}(u,p)$, by examining both terms of the $\max$ expression in Eq. 8.

i) Since h is Δ-consistent, we get that $h(u,v) \leq h(u,p) + h(p,v)$ for all $u,p,v \in V$. Thus, $h(u,v) - h(p,v) \leq h(u,p)$, proving the lemma for the first term.

ii) The triangle inequality from v to p (where u is a pivot) gives: $h(v,p) \leq h(v,u) + h(u,p)$ for all $u,p,v \in V$. Thus, $h(v,p) - h(v,u) \leq h(u,p)$, proving the lemma for the second term. □

Therefore, if h is Δ-consistent, there is no benefit in using $\hat{h}$ since it will never exceed the value of h. A similar proof applies to the case of undirected graphs where for a Δ-consistent heuristic h, we have that:

1. $\hat{h}(u,v) = |h(u,p) - h(p,v)| \leq d(u,v)$. This is due to the consistency attribute.
2. $\hat{h}(u,v) = |h(u,p) - h(p,v)| \leq h(u,v)$. This is due to the Δ-consistency attribute.

We note that computing h_{HDH} can be computationally expansive, as the right-hand side of Eq. 8 depends on all vertices in the graph. Furthermore, a common assumption in search algorithms is that they only have access to the graph through state expansions. Under this constraint, computing h_{HDH} by taking the maximum over all vertices of G becomes infeasible. However, it is feasible to approximate h_{HDH} by considering only the vertices discovered during the search process, thereby reducing computational complexity while maintaining the aforementioned expansion assumption. Alternatively, one can choose a fixed set of vertices V' and compute the right-hand side of Eq. 8 while only considering pivots from V'.

Demonstration of HDH on a Search Problem

If h is consistent but not Δ-consistent, Lemma 2 does not apply, and there might be states n, g where $\hat{h}(n,g) > h(n,g)$. Thus, with a consistent heuristic A* can use h_{HDH} as an enhanced heuristic to potentially improve its performance.

Consider again the graph Figure 2(a) and its heuristic values in Figure 2(b). We have shown above that this heuristic is admissible and consistent but that it does not obey the triangle inequality. We now focus on $h(u,p) = 1$ and are interested in improving it by calculating $h_{\text{HDH}}(u,p)$. The first term in the max gives $h(u,v) - h(p,v) = 6 - 3 = 3$, while the second term gives $h(v,p) - h(v,u) = 10 - 2 = 8$, resulting in $\hat{h}(u,p) = 8$. This implies that $h_{\text{HDH}}(u,p) = max(1,8) = 8$.

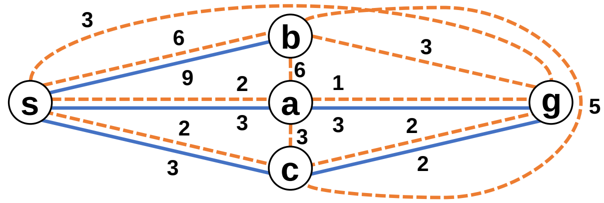


Figure 3: Example: enhancing a weakly-consistent heuristic.

Consider the triangle inequality in the form $h(x, y) \leq h(x, z) + h(z, y)$. It means that the direct path from x to y is no worse than the bypass from x to y via z, denoted $(x \rightarrow z \rightarrow y)$. Consider edge (u, p) in Figure 2. It can be part of a bypass in two cases. The first case considers paths from u to v. In this case, (u, p) is first in the bypass $(u \rightarrow p \rightarrow v)$. Here $h(u, v) = 6$ while the bypass path via p gives $h(u, p) + h(p, v) = 1 + 3 = 4$. Clearly, the triangle inequality does not hold in this case. To rectify this, we can set $h(u, p) = 3$, as done in the first term in $\hat{h}(u, p)$, ensuring that the bypass distance is at least 6, as dictated by the triangle inequality. The second case considers paths from v to p. In this case, edge (u, p) is second in the bypass $(v \rightarrow u \rightarrow p)$. Here the direct edge gives $h(v, p) = 10$ while the bypass path via u gives $h(v, u) + h(u, p) = 2 + 1 = 3$. Clearly, the triangle inequality also does not hold in this scenario. To address this, we can set $h(u, p) = 8$, ensuring that the bypass distance is at least 10, as required by the triangle inequality. This adjustment is implemented through the second term in the formula of $\hat{h}(u, p)$.

Next, consider the undirected graph in Figure 3, in which solid blue lines indicate the edge costs and dashed orange lines indicate heuristic values, and focus on the path from s to g. In this example, the heuristic is consistent, but not Δ-consistent. While $h(a, g) = 1$, observe that $\hat{h}(a, g) = |h(a, b) - h(g, b)| = 6 - 3 = 3$. This information can be utilized by A*. In particular, in this example, using only the original heuristic h, A* would expand s with $f(s) = 3$, followed by a with $f(a) = g(a) + h(h) = 3 + 1 = 4$. Then it will expand c and g (both with $f = 5$) and find the optimal solution. By contrast, using $\hat{h}$, $f(a) = g(a) + \hat{h}(a) = 3 + 3 = 6$. Now, A* does not expand a and directly expands c and g. Notably, this example can be modified such that A* would need to expand an arbitrarily large number of states when not using h_{HDH}. Moreover, the computation of $\hat{h}$ in this instance only entails the goal and states expanded during the search, thereby upholding the assumption that states are accessible solely through expansions.

Importantly, the proof of optimal efficiency for A* (Dechter and Pearl 1985) hinges on the assumption that the algorithm uses a goal-specific heuristic. To the best of our knowledge, the optimality of A* has not been explored in the context of all-pair heuristics. However, the provided example illustrates a significant point: when A* receives an all-pair heuristic and treats it as a goal-specific heuristic, as often practiced, it may not achieve optimal efficiency compared to algorithms capable of leveraging pairwise heuristic estimations, particularly when the heuristic lacks Δ-consistency. It remains an open question whether A* is optimally efficient equipped with an all-pair Δ-consistent heuristic, a matter we leave for future work.

Characterizing Common Heuristic Families

We will now explore common families of heuristic functions and show that they adhere to the triangle inequality.

An *abstraction transformation* for graph $G = (V, E)$ and a cost function c can be defined as follows: Let $G' = (V', E')$ be a graph, $c' : E' \rightarrow \mathbb{R}^{\geq 0}$ be a cost function on G', and $A : V \rightarrow V'$ be a function that maps vertices of G to vertices of G'. We denote by *all-pair abstraction-based heuristic* a heuristic function $h_{(G', A, c')} : V \times V \rightarrow \mathbb{R}^{\geq 0}$ of G, where the heuristic value between two states is the cost of the shortest paths between them when mapped into G'. That is, $h_{(G', A, c')}(u, v) = d_{G'}(A(u), A(v))$, where $d_{G'}(A(u), A(v))$ denotes the shortest path in G' between $A(u)$ and $A(v)$, w.r.t. c'.

All-pair abstraction-based heuristics encompass various commonly used techniques, such as constraint relaxation (where edges are added) (Gaschnig 1979; Hansson, Mayer, and Yung 1992; Bonet and Geffner 2001), domain abstractions (Hernádvölgyi and Holte 2000; Kreft et al. 2023), STAR abstractions (Holte et al. 1996; Botea, Müller, and Schaeffer 2004; Sturtevant and Buro 2005), and homomorphism abstraction, which includes pattern databases (PDBs) (Kibler 1982; Culberson and Schaeffer 1998; Felner, Korf, and Hanan 2004), Merge-and-shrink (Dräger, Finkbeiner, and Podelski 2009; Helmert et al. 2014; Sievers and Helmert 2021), and other techniques.

Lemma 3. *All-pair abstraction-based heuristics obey the triangle inequality.*

Proof. Assume by contradiction that an all-pair abstraction-based heuristic h does not hold the triangle inequality. Thus, there exists three states, u, v, p such that $h(u, v) > h(u, p) + h(p, v)$. By definition, $h(n, m) = d_{G'}(A(n), A(m))$ for all states $n, m \in V$. Hence, $d_{G'}(A(u), A(v)) > d_{G'}(A(u), A(p)) + d_{G'}(A(p), A(v))$. Consequently, $d_{G'}$ does not satisfy the triangle inequality in G', in contradiction to the fact that the cost of the shortest paths between vertices in any graph is quasi-pseudo-metric. □

Additionally, when a set of heuristics adheres to the triangle inequality, both their sum and maximum also preserve this property.

Lemma 4. *Heuristics conforming to the triangle inequality maintain this property under the SUM and MAX operations.*

Proof. Let h_1 and h_2 be any two heuristics that obey the triangle inequality, $u, v, p \in V$ be any three states, $h_S = h_1 + h_2$, and $h_M = \max(h_1, h_2)$. Since the triangle inequality holds for h_1 and h_2, $h_1(u, v) \leq h_1(u, p) + h_1(p, v)$ and $h_2(u, v) \leq h_2(u, p) + h_2(p, v)$. By summing these two inequalities, we get that $h_1(u, v) + h_2(u, v) \leq h_1(u, p) + h_2(u, p) + h_1(p, v) + h_2(p, v)$. By replacing $h_1 + h_2$ with h_S we get: $h_S(u, v) \leq h_S(u, p) + h_S(p, v)$. Consequently,

h_S obeys the triangle inequality, showing that this property is maintained under summation.

In addition, assume w.l.o.g. that $h_1(u,v) \leq h_2(u,v)$ for a pair of states u,v. Since $h_2(u,v) \leq h_2(u,p) + h_2(p,v)$, it follows that $\max(h_1(u,v), h_2(u,v)) \leq h_2(u,p) + h_2(p,v)$. Moreover, given that $h_2(n,m) \leq \max(h_1(n,m), h_2(n,m))$ for all $n,m \in V$, we have $\max(h_1(u,v), h_2(u,v)) \leq \max(h_1(u,p), h_2(u,p)) + \max(h_1(p,v), h_2(p,v))$. By replacing $\max(h_1, h_2)$ with h_M we obtain $h_M(u,v) \leq h_M(u,p) + h_M(p,v)$. Hence, h_M adheres to the triangle inequality, demonstrating that this property is preserved under maximization. □

Since all-pair abstraction-based heuristics satisfy the triangle inequality, Lemma 4 implies that the summation and maximization of such heuristics also maintain this property. These operations encompass various heuristic techniques, including additive PDBs and selecting the maximum value among different PDBs. Notably, despite trying, the authors of this paper cannot conceive of a non-synthetic consistent and admissible all-pair heuristic that violates the triangle inequality. This remains a challenge for the entire community.

Conclusions

This paper studies *all-pair consistent heuristic function* and their relation to the *all-pair shortest path function* with regard to different distance properties. We defined a new property of Δ-consistency, a special case of consistency in which the heuristic adheres to the triangle inequality with respect to the states in the graph. Moreover, we introduced HDH, a novel method for extending the triangle inequality to consistent heuristics, making them Δ-consistent. We then proved that most existing heuristics already maintain the triangle inequality and are thus Δ-consistent. Two intriguing questions remain unanswered:
i) Are there natural mechanisms for constructing heuristics that result in heuristics that are consistent but not Δ-consistent?
ii) Is A* optimally efficient when given a Δ-consistent all pair heuristic?
These inquiries are left as challenges for the search community to explore in the future.

Acknowledgments

This work was supported by the Israel Science Foundation (ISF) grant #909/23 awarded to Shahaf Shperberg and Ariel Felner, by Israel's Ministry of Innovation, Science and Technology (MOST) grant #1001706842, awarded to Shahaf Shperberg, and by United States-Israel Binational Science Foundation (BSF) grant #2021643 awarded to Ariel Felner. This work was also partially funded by the Canada CIFAR AI Chairs Program. We acknowledge the support of the National Sciences and Engineering Research Council of Canada (NSERC).

References

Bonet, B.; and Geffner, H. 2001. Planning as heuristic search. *Artif. Intell.*, 129(1-2): 5–33.

Botea, A.; Müller, M.; and Schaeffer, J. 2004. Near Optimal Hierarchical Path-Finding. *J. Game Dev.*, 1(1): 1–30.

Chartrand, G.; and Tian, S. 1997. Distance in digraphs. *Computers & Mathematics with Applications*, 34(11): 15–23.

Culberson, J. C.; and Schaeffer, J. 1998. Pattern Databases. *Comput. Intell.*, 14(3): 318–334.

de Champeaux, D.; and Sint, L. 1977. An improved bidirectional heuristic search algorithm. *Journal of the ACM (JACM)*, 24(2): 177–191.

Dechter, R.; and Pearl, J. 1985. Generalized Best-First Search Strategies and the Optimality of A*. *J. ACM*, 32(3): 505–536.

Dräger, K.; Finkbeiner, B.; and Podelski, A. 2009. Directed model checking with distance-preserving abstractions. *Int. J. Softw. Tools Technol. Transf.*, 11(1): 27–37.

Eckerle, J.; Chen, J.; Sturtevant, N. R.; Zilles, S.; and Holte, R. C. 2017. Sufficient Conditions for Node Expansion in Bidirectional Heuristic Search. In *Proceedings of the Twenty-Seventh International Conference on Automated Planning and Scheduling, ICAPS*, 79–87.

Felner, A.; Korf, R. E.; and Hanan, S. 2004. Additive Pattern Database Heuristics. *J. Artif. Intell. Res.*, 22: 279–318.

Felner, A.; Zahavi, U.; Holte, R.; Schaeffer, J.; Sturtevant, N. R.; and Zhang, Z. 2011. Inconsistent heuristics in theory and practice. *Artif. Intell.*, 175(9-10): 1570–1603.

Gaschnig, J. 1979. A Problem Similarity Approach to Devising Heuristics: First Results. In *Proceedings of the Sixth International Joint Conference on Artificial Intelligence, IJCAI 79, Tokyo, Japan, August 20-23, 1979, 2 Volumes*, 301–307. William Kaufmann.

Hansson, O.; Mayer, A.; and Yung, M. 1992. Criticizing solutions to relaxed models yields powerful admissible heuristics. *Inf. Sci.*, 63(3): 207–227.

Hart, P. E.; Nilsson, N. J.; and Raphael, B. 1968. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Trans. Syst. Sci. Cybern.*, 4(2): 100–107.

Helmert, M.; Haslum, P.; Hoffmann, J.; and Nissim, R. 2014. Merge-and-Shrink Abstraction: A Method for Generating Lower Bounds in Factored State Spaces. *J. ACM*, 61(3): 1–63.

Hernádvölgyi, I. T.; and Holte, R. C. 2000. Experiments with Automatically Created Memory-Based Heuristics. In *Abstraction, Reformulation, and Approximation, 4th International Symposium, SARA 2000, Horseshoe Bay, Texas, USA, July 26-29, 2000, Proceedings*, volume 1864 of *Lecture Notes in Computer Science*, 281–290. Springer.

Holte, R. C.; Mkadmi, T.; Zimmer, R. M.; and MacDonald, A. J. 1996. Speeding up Problem Solving by Abstraction: A Graph Oriented Approach. *Artif. Intell.*, 85(1-2): 321–361.

Kaindl, H.; and Kainz, G. 1997. Bidirectional heuristic search reconsidered. *Journal of Artificial Intelligence Research*, 7: 283–317.

Kibler, D. 1982. Natural generation of admissible heuristics. In *UC Irvine: Donald Bren School of Information and Computer Sciences*, TR-188.

Kreft, R.; Büchner, C.; Sievers, S.; and Helmert, M. 2023. Computing Domain Abstractions for Optimal Classical Planning with Counterexample-Guided Abstraction Refinement. In *Proceedings of the Thirty-Third International Conference on Automated Planning and Scheduling, July 8-13, 2023, Prague, Czech Republic*, 221–226.

Siag, L.; Shperberg, S.; Felner, A.; and Sturtevant, N. R. 2023. Comparing Front-to-Front and Front-to-End Heuristics in Bidirectional Search. In *Proceedings of the International Symposium on Combinatorial Search, SoCS*, volume 16, 158–162.

Sievers, S.; and Helmert, M. 2021. Merge-and-Shrink: A Compositional Theory of Transformations of Factored Transition Systems. *J. Artif. Intell. Res.*, 71: 781–883.

Sturtevant, N. R.; and Buro, M. 2005. Partial Pathfinding Using Map Abstraction and Refinement. In *Proceedings, The Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference, July 9-13, 2005, Pittsburgh, Pennsylvania, USA*, 1392–1397.

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