Of Random Beacons and Heuristics

Algorithmic Machine Learning and Data Science
Fall 2023 - Final Project Report

Pratyush Avi
pa2439@nyu.edu

Ryan Kim
rk2546@nyu.edu

Abstract

The performance of the A path-finding algorithm can be represented as* "additive overhead"*, or the number of nodes not on any shortest path on a given weighted graph that happened to be visited by A* during the query process. The lower bound on the expected additive overhead of A* is well-defined for two types of heuristic functions: Norm and Labeling Heuristics. However, the upper bound proof is not as rigorously constructed or well-defined as its lower bound counterparts. This report attempts to elucidate missing details in the upper bound proof sketch for the Beacon Heuristic, an extension of the Norm Heuristic, and outline potential areas of exploration with this new heuristic method.*

1. Introduction

The paper, *"Embeddings and Labeling Schemes for A*"*, published by Eden, Indyk, and Xu in 2021 [4], took inspiration from existing literature [2, 3, 6] to formalize a a study regarding the complexity of A* search using "learned heuristics". Learned heuristics are generated by predictors that estimate the distance between two nodes based on the features of nodes, with such features being either manually determined by engineers and researchers or precomputed using an additional pre-processing step.

The performance of the A* path-finding algorithm can be represented as *"additive overhead"*, or the number of nodes not on any shortest path on a given weighted graph that happened to be visited by A* during the query process. The lower and upper bounds of this additive overhead will depend greatly on the type of heuristic function used with the A* implementation, the type of graph used, and the dimensionality of the embeddings generated for each node in the graph. To account for this, this paper relies on global assumptions regarding the nature of the graphs in question and the performance of the A* algorithm such as whether the graphs are weighted and that ties are broken arbitrarily and efficiently with no significant contribution to runtime complexity. Furthermore, the paper generalizes the heuristic function types outputted by predictors into two broad categories: the Norm Heuristic and the Labeling Heuristic, with a subtype called the Beacon Heuristic extending from the Norm Heuristic.

The lower bounds on additive overhead are proven rigorously with regard to Norm and Labeling Heuristics, with their proof sketches showcasing elaborate depictions of possible graph scenarios and how these scenarios inform us of the additive overhead generated from either the Norm or Labeling Heuristic. The upper bound proof for this additive overhead, spearheaded by the Beacon Heuristic, leaves us with more questions regarding its admissibility and further extensions of it as a legitimate heuristic function.

This report therefore is motivated by a desire to better understand the exact behavior of the Beacon Heuristic and how the upper bound proof sketch is guaranteed by this behavior. Insofar as this report's authors are concerned, establishing a clearer definition of the upper bound proof was the primary goal. This report summarizes the upper bound proof for the Beacon Heuristic and highlights details in the proof that yet remain elusive in its meaning and require further elaboration in potential future works.

2. About the Paper: A Review

2.1. Defining Complexity

Eden, Indyk, and Xu quantify the impact of a particular heuristic on A* search through the idea of *"additive overhead"*: the number of additional nodes **not** on the true shortest path that the A* algorithm will visit during the query process. Formally, for a weighted graph $G = (V, E, W)$ and a heuristic function h, let $P(s,t)$ be the set of vertices on the shortest path between vertices s and t. Let h be a heuristic function, and let $S_h(s,t)$ be the set of vertices scanned by A* given s,t,h. The paper declares that heuristic h will have some additive overhead c **on average** if:

$$\mathbb{E}_{s,t \in V}\left[|S_h(s,t)| - |P(s,t)|\right] \leqslant c$$

where s and t are chosen independently and uniformly at random from V (hence the expected value notation). In other words, the number of additional vertices visited by A* will never go above some constant c.

2.2. Heuristic Functions

There are two key assumptions regarding the heuristic function that are emphasized by the original paper:

- Every heuristic function explored in the the original paper is *consistent* or *monotonic* [5].
- Every heuristic function explored is also *admissible* - that the heuristic function will never overestimate the true distance between two vertices. While admissibility is assumed if a heuristic is already consistent, it is important to emphasize the admissibility assumption.

A heuristic function h is consistent if it satisfies the following:

$$h(x) \leq dist(x,y) + h(y)$$

for every edge (x,y) of the graph and where $dist(x,y)$ represents the true length of the edge. To put it in another way, if x and y are two connected vertices in a graph, then the true edge length between x and y added to the heuristic estimate for the distance between y and the end destination vertex must be greater than or equal to the heuristic estimate of x to the destination vertex. If a heuristic function is known to be consistent, then A* is guaranteed to find an optimal path without processing any node more than once; you'll never need to revisit any nodes when you're searching your graph [5].

2.3. Tie-Breaks in A*

An additional assumption key to the Beacon Heuristic in particular is the reliance on tie-breaks: the A* algorithm's is ability to break ties arbitrarily and efficiently. A formal declaration of this assumption is pulled from the paper:

> **Assumption 1.9** (Breaking ties): *We assume that, when there is a tie on the estimated distance lower bound (as specified in* Definition 2.3*) for multiple different vertices, the A* algorithm chooses the next vertex in a way that minimizes the total running time.*

The Definition 2.3 mentioned in the assumption is as follows:

> **Definition 2.3** (Estimated distance lower bound): *Given a query pair $dist(s,t)$, for any vertex u, we define $g(u)$ to be the estimated distance lower bound calculated by A* for shortest path from s to t passing u: $g(u) = dist(s,u) + h(u,t)$. In particular, $g(t) = dist(s,t)$.*

2.4. Heuristic Types and Bounds

The paper looks at three generalized forms of heuristic functions that may appear when using learned heuristics: the Norm Heuristic, the Labeling Heuristic, and the Beacon Heuristic (which is an extension of the Norm Heuristic). The formal definitions of these heuristics are provided below and are taken directly from the original paper:

> **Norm Heuristic**: *A heuristic function $h : V \times V \to \mathbb{R}$ is a Norm Heuristic induced by another function $\pi : V \to \mathbb{R}^d$ and the l_p norm if $h(s,t) = \|\pi(s) - \pi(t)\|_p$ for any pair of vertices $s,t \in V$. The dimension d is referred to as the dimension of h.*

Labeling Heuristic: *A heuristic function* $h : V \times V \to \{0, \ldots, C\}$ *is a Labeling Heuristic induced by functions* $f : V \to \{0, \ldots C\}^L$ *and* $g : \{0, \ldots, C\}^L \times \{0, \ldots, C\}^L \to \{0, \ldots, C\}$ *if* $h(s,t) = g(f(s), f(t))$ *for any pair of vertices* $s, t \in V$. *Here the parameter* C *is assumed to be polynomial in* $|V|$ *and* L *is referred to as the label length.*

Beacon Heuristic: *A heuristic function* $h : V \times V \to \mathbb{R}$ *is a Beacon Heuristic induced by set* $B \subset V$ *if* $h(s,t) = \|\pi(s) - \pi(t)\|_\infty$ *where* $\pi(s) = (dist(s, b_1), \ldots, dist(s, b_{|B|}))$ *(so that* $h(s,t) = max_{b \in B} |dist(s,b) - dist(t,b)|$*)*

The findings for the upper and lower bounds of each heuristic type are described in Figure 1. Note that each bound relies on some determination of d, or the dimensionality of the vertex embeddings. Each proof for these determinations relies on some combination of setting and assumptions, represented by the "Settings" and "Comments" columns of the table. For example, the upper bound of the Beacon Heuristic relies on the assumption that all tie breaks can be broken efficiently (**Assumption 1.9**), and some lower bound proofs for the Norm Heuristics differentiate between unweighted and weighted graphs.

3. Beacon Heuristic

Observe in Figure 1 that the upper bound for the additive overhead for A* with regard to both the Norm and Labeling Heuristic types ends up being $O(n^{1-\alpha})$ in expectation (on average) for an embedding size of $O(n^\alpha)$. However, a cursory glance at the original report shows that the proofs of the upper bounds with regard to the Beacon and Labeling Heuristics are not as clearly defined as the lower bounds for the Norm and Labeling Heuristics. Therefore, the rest of this report will discuss the Beacon Heuristic in isolation and will not refer to either the Norm or the Labeling Heuristics separately.

3.1. About the Beacon Heuristic

The Beacon Heuristic is a special variant of the l_∞-Norm Heuristic which is induced by a set of "beacons". These beacons are a set of randomly selected nodes from the set of vertices, i.e., $B \subset V$. For a source vertex u and destination vertex v, the Beacon Heuristic first computes the length of the path from u to v through every beacon in B and returns the largest distance. We are assuming these distances are some form of Euclidean or Manhattan distances and do not actually make a comment about the connections between each node and the beacons. Formally, we can describe the Beacon Heuristic as follows:

Beacon Heuristic: *A function* $h : V \times V \to \mathbb{R}$ *induced by a set* $B \subset V$ *s.t.,* $h(s,t) = \|\pi(s) - \pi(t)\|_\infty$, *where* $\pi(s) = (dist(s, b_1), ..., dist(s, b_{|B|})), \forall b \in B$

The l_∞-norm selects the largest value in a sequence. Therefore, we can rewrite the Beacon Heuristic as follows:

Beacon Heuristic: *A function* $h : V \times V \to \mathbb{R}$ *induced by a set* $B \subset V$ *s.t.,* $h(s,t) = max_{b \in B} |dist(s,b) - dist(t,b)|$

The intuition behind this is pretty straightforward: in the worst case, we want to judge a potential node based on its worst estimate. If we were to choose a $node$ from a set of nodes U to get to a destination t, we would choose the one that returns the least worst estimate, i.e.,

$$\begin{aligned} node &= \text{argmin}_{u \in U} h(u,t) \\ &= \text{argmin}_{u \in U} (\max_{b \in B} |dist(u,b) - dist(t,b)|) \end{aligned}$$

A potential vertex may have the lowest cost for a certain beacon but also have a terrible cost for another. Since we have no guarantees about whether that first beacon will lie on the shortest path, we have to assume that the beacon with largest distance can lie on the path of this node in the worst case. This stems from the idea that the Beacon Heuristic does not actually compute the actual path length between any two nodes in the graph (that would be another shortest-path search onto itself!!) but simply returns the distance between their embeddings or labels.

3.2. Addendum

The Beacon Heuristic appears to have been inspired by a separate paper by Abraham et al. [1]. However,

<table>
<tr><th>Type</th><th>Settings</th><th>Comments</th><th>Result</th><th>Reference</th></tr>
<tr><td rowspan="5">Norm heuristics</td><td>$p < \infty$</td><td rowspan="2"></td><td>$d = o(n/\log n) \Rightarrow \tilde{\Omega}(n)$ avg. overhead</td><td rowspan="2">Thm. 1.6</td></tr>
<tr><td>$p = \infty$</td><td>$d = o(\log n) \Rightarrow \tilde{\Omega}(n)$ avg. overhead</td></tr>
<tr><td rowspan="3">$p = \infty$
weighted graphs</td><td>general graphs</td><td rowspan="2">$d = o(n^{\alpha}) \Rightarrow \tilde{\Omega}(n^{1-\alpha})$ avg. overhead</td><td>Thm. 1.7</td></tr>
<tr><td>grid graphs ($\alpha < 0.5$)</td><td>Thm. 3.4</td></tr>
<tr><td>beacon based
Assumption 1.9</td><td>$d = O(n^{\alpha})$ and $O(n^{1-\alpha})$ avg. overhead</td><td>Thm. 1.10</td></tr>
<tr><td rowspan="3">Labeling heuristics</td><td rowspan="3">weighted graphs</td><td>general graphs</td><td rowspan="2">$L = o(n^{\alpha}) \Rightarrow \Omega(n^{1-\alpha})$ avg. overhead</td><td>Thm. 1.8</td></tr>
<tr><td>grid graphs ($\alpha < 0.5$)</td><td>Thm. 3.9</td></tr>
<tr><td>Assumption 1.11</td><td>$L = O(n^{\alpha})$ and $O(n^{1-\alpha})$ avg. overhead</td><td>Thm. 1.12</td></tr>
</table>

Figure 1. the findings regarding the upper and lower bounds of each heuristic form. Lower bounds are colored in white, while upper bounds are colored in gray

upon literary analysis of Abraham et al.'s paper, there appears to be little to no correlation between the Beacon Heuristic defined here and the concepts explored by Abraham et al. in their paper. Abraham et al. were particularly focused on trying to solve whether sublinear query bounds for a certain class of heuristic functions based on *"contraction hierarchies"* was possible and identify what properties of real-world highway networks allow these heuristics to perform so well in practice. A contraction hierarchy is a method of graph preprocessing where shortcuts are identified between nodes that are guaranteed to be connected via extraneous nodes. During this shortcut identification process, a "hierarchy" controls the order of nodes that must be pre-processed, as the number and directionality of shortcuts in a graph will be highly dependent on which nodes are explored first. When shortcuts are identified, any pathfinding algorithm will therefore be able to operate on a truncated version of the original graph that is made primarily of shortcuts, thus optimizing on query time. Abraham et al. then make a point to highlight heuristic algorithms that are reliant on contraction hierarchies as a means of graph navigation. Therefore, given that different graphs will face varying levels of success in identification of shortcuts, Abraham et al. derive a quality about highway networks called the **highway dimension** that drive a graph's ability to be pre-processed and queried using contraction hierarchies. For what Abraham's paper is worth, there is little to no connection with the Beacon Heuristic defined in this paper. The lack of connection prevents any robust analysis of the Beacon Heuristic in practice, as no other papers have been identified to use the Beacon Heuristic.

3.3. Primers for the Upper Bound Proof

As highlighted in the introduction, several key assumptions are made about the nature of A* that are key for the upper bound additive cost proof.

- The upper bound proof relies on the idea that the graph is a weighted graph. There is no mention of how the upper bound performs with unweighted graphs.
- The performance of the Beacon Heuristic is measured with **Assumption 1.9** in mind - that the system efficiently and arbitrarily breaks ties. This tie-break assumption affects this particular fact mentioned in the original paper:

> **Fact 2.4** (Scan condition, see [5]): *Suppose we apply A* to calculate $dist(s,t)$ with a consistent heuristic function $h(u,t)$. Then a vertex u must be scanned if and only if $dist(s,u)+h(u,t) < dist(s,t)$ and may be scanned if and only if $dist(s,u) + h(u,t) \leqslant dist(s,t)$*

These two primers mean that when answering a query $dist(s,t)$, A* will only scan a vertex u if either:

- $u \in P(s,t)$
- $dist(s,u) + h(u,t) < dist(s,t)$, or equivalently $h(u,t) < dist(s,t) - dist(s,u)$

3.4. The Upper Bound Proof

The full definition of the beacon-based upper bound is as follows, extracted from the original paper verbatim:

> **Theorem 1.10** (Beacon-based upper bound): *Under Assumption 1.9, for any weighted graph $G = (V, E, W)$ with $|V| = n$ vertices, there exists a Beacon Heuristic induced by a set B with $|B| = n^{\alpha}$ with an additive overhead $n^{1-\alpha}$, i.e., such that A* scans at most $n^{1-\alpha} + E_{s,t}[|P(s,t)|]$ vertices on average.*

The first part of this proof therefore starts off relatively simple:

1. Select a uniformly random beacon set B from the vertex set V.
2. Uniformly and randomly select a pair of vertices s, t that act as the start and destination nodes for an A* query.
3. Based on our pre-established primers, an A* query for (s,t) will visit a node u not on the shortest path $P(s,t)$ if $h(u,t) < dist(s,t) - dist(s,u)$,
4. By definition of the Beacon Heuristic, $h(u,t) = \|\pi(u) - \pi(t)\| = max_{b \in B}|dist(u,b) - dist(b,t)|$. In other words, $h(u,t)$ is defined as the maximum possible distance between u and t given that the path passes through a beacon node.

After this step, the proof diverts to the perspective of node u, where all the beacons in B will be compared to the starting node s. The steps are as follows, in this case taken from the paper almost verbatim:

5. "u will satisfy the above only if s can give a better distance estimation for $dist(u,t)$ than all beacons in B."
6. "The above will occur with probability at most $\frac{1}{|B|+1}$ over a random selection of $B \cup \{s\}$ for a fixed pair (u,t)."
7. "In expectation, for any pair (u,t), there are at most $\frac{n}{|B|}$ such vertices s."
8. "Therefore, taking a summation over all pairs of (u,t) yields that the expected number of extra vertices scanned is at most $\frac{n^2 \cdot \frac{n}{|B|}}{n^2} = \frac{n}{|B|}$."
9. "Finally, we choose $|B| = n^{\alpha}$ to get the desired result."

3.5. Barriers to Understanding

3.5.1 "Better estimation"

Steps 5 and 6 see that we include the start node s as a possible consideration for the heuristic distance prediction $h(u,t)$. In other words, we would consider $h(u,t) = max_{v \in B \cup \{s\}}|dist(u,v) - dist(v,t)|$ as opposed to just the beacon nodes. This concept is strengthened by Step 6, where $|B|+1$ is in reference to the number of nodes in B plus the start node s. However, the term "better estimation" is up to interpretation in this regard. Based on the authors' perception of this proof, there are two major interpretations that are viable:

- The term "better" refers to how $h(u,t)$ more closely approximates $dist(u,t)$ in that, if we assume a path (not the shortest path) exists from s to t through u, that the heuristic equation gives just that: $h(u,t) \approx dist(u,s) - dist(s,t) = dist(u,t)$.
- The term "better" refers to how, when compared to all other beacons in B, the start node s is furthest away from all other beacons which consequently happen to be close to u in terms of distance. This "better" is optimal for u, as it implies based on the definition for the heuristic that moving through u might be a "good" option for A* to consider; there is a path (u,b,t) that might be more optimal to consider than if one were to move through the another path.

3.5.2 "In expectation, for any pair..."

Step 7 sees that we expand the idea of the probability $\frac{1}{|B|+1}$ of s giving a "better" estimation than all other beacons to that of all potential s nodes for any node pair (u, t). They state that the expected number of such vertices s is defined as, at most, $\frac{n}{|B|}$. Here, s now refers to any arbitrary node that could at as the start point for a path $P(s, t)$. We can therefore intuit that this number $\frac{n}{|B|}$ is an upper bound on the number of such s vertices, or n_s:

$$n_s = \sum_{i=1}^{n} Pr(v_i \text{ gives better estimate})$$

In expectation, this becomes:

$$E[n_s] = E\left[\sum_{i=1}^{n} Pr(v_i \text{ gives better estimate})\right]$$

Due to linearity of expectation, we can further reduce this expression down to the following:

$$\begin{aligned} E[n_s] &= \sum_{i=1}^{n} E\left[v_i \text{ gives a better estimate}\right] \\ &= \sum_{i=1}^{n} \frac{1}{|B|+1} \\ &= \frac{n}{|B|+1} \\ &< \frac{n}{|B|} \end{aligned}$$

3.5.3 "Summation over all pairs..."

Step 8 regards that the expected number of extra vertices scanned during an A* query is equivalent to $\frac{n^2 \cdot \frac{n}{|B|}}{n^2} = \frac{n}{|B|}$. This step transforms the perspective from u back to A*, such that rather than count how many s vertices give a good estimate of $dist(u, t)$ for all pairs (u, t), we count how many vertices u will be visited by $A*$ starting from any start vertex s. The language of the proof implies some union bound is required for this next step, but beyond this the mathematical proof is lacking in its explicit definition.

4. Discussion

Unclear elements in the upper bound proof, as well as the nature for how heuristics are defined in this paper, may open or restrict potential expansions or empirical analysis of the Beacon Heuristic. Recall that the Norm, Labeling, and Beacon Heuristics are all based on **learned** embeddings from some trained predictor. This means that any attempt to recreate these conditions manually (i.e. generating an Erdős–Rényi graph to replicate a potential 2D embedding space for a set of sparsely connected nodes) may not necessarily reflect the actual properties of the predictors' embeddings. This raises some concerns regarding an empirical analysis of any of the heuristics provided in this paper.

One topic of discussion was whether the Beacon Heuristic in its definition is actually admissible, given that it involves finding the maximum distances of beacons away from a query node u. The proof relies on the assumption that all heuristics explored in the original paper are both consistent and admissible, but given how explicitly the beacon heuristic is defined, it opens the discussion as to whether the assumption of admissibility is valid.

Outside of the Beacon Heuristic, the lexicon and terminology employed by Eden, Indyk, and Xu throughout the paper is ITCS-heavy and is not immediately understandable by the general layman. While this is naturally expected for a publication of this caliber, it was unexpected that certain variable labels and terms were sometimes flipped, such as d and $dist$ sometimes used interchangeably while d also refers to the dimensionality of the embeddings for each node.

4.1. Further Exploration

The Beacon Heuristic is perhaps the most interesting heuristic defined in the original paper given its explicit definition and the unique manner in which it attempts to provide a heuristic estimate for the distances between nodes. One area of inquiry visited by this report's authors is how the presence of fully-connected cliques would have affected the performance of the Beacon Heuristic. This is entirely based on the concept that perhaps A* could be "trapped" within a clique if one or more beacons were randomly determined to be contained within said clique, or how the heuristic's performance may fail if either the start or destination nodes of an A* query happened to end up in one or more cliques. The assumption made by this reports authors were that, if cliques contributed to the additive overhead of the Beacon Heuristic, then a new upper bound for the Beacon Heuristic for this subclass of graphs could potentially be identified.

5. Conclusion

The analysis of the Beacon Heuristic and its upper bound proof highlight areas that require further elaboration in order for any theoretical expansion such as tighter/looser upper bounds to be possible. This suggests that a potential addendum and expansion of the upper bound proof that more explicitly explain the steps to achieve $O(n^{1-\alpha})$ are necessary. This report attempts to perform that elucidation as a starting point. Furthermore, the original paper inspires new graph characteristics that may, in theory, push the upper bound of the Beacon Heuristic further than $O(n^{1-\alpha})$.

References

[1] Ittai Abraham, Amos Fiat, Andrew V. Goldberg, and Renato F. Werneck. Highway dimension, shortest paths, and provably efficient algorithms. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics, Jan. 2010. 3

[2] Mohak Bhardwaj, Sanjiban Choudhury, and Sebastian Scherer. Learning heuristic search via imitation. In Sergey Levine, Vincent Vanhoucke, and Ken Goldberg, editors, *Proceedings of the 1st Annual Conference on Robot Learning*, volume 78 of *Proceedings of Machine Learning Research*, pages 271–280. PMLR, 13–15 Nov 2017. 1

[3] Binghong Chen, Chengtao Li, Hanjun Dai, and Le Song. Retro*: Learning retrosynthetic planning with neural guided a* search. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1608–1616. PMLR, 13–18 Jul 2020. 1

[4] Talya Eden, Piotr Indyk, and Haike Xu. Embeddings and labeling schemes for a*, 2021. 1

[5] Peter Hart, Nils Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968. 2, 4

[6] Ryo Yonetani, Tatsunori Taniai, Mohammadamin Barekatain, Mai Nishimura, and Asako Kanezaki. Path planning using neural a* search. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12029–12039. PMLR, 18–24 Jul 2021. 1