Extended Abstract: Topological Analysis of Admissible Heuristics in IDA*

I. Goal
Given an admissible heuristic, a problem instance and an informed search based problem solver we want to predict accurately the size of the search tree generated to find an optimum solution.

II. Introduction
Heuristics are created to reduce the time it takes to solve problems. Admissible heuristics are used to perform informed searches (A*,IDA*,etc) to find optimal solutions to problems.

Figure 1: Informed Search Diagram

n is the current state.
g(n) is the Optimal Distance (OD) from I to n
h(n) is the OD from n to G
\( \hat{h} \) is the heuristic estimate OD from n to G.
F(n)=g(n)+h(n)=OD.

\( \hat{F}(n)=g(n)+\hat{h}(n) \) = estimated optimal distance

Each dot represents a problem state in the optimal path from initial state I to goal state G. Informed search on planning expands all nodes from I to G whose \( \hat{F}(n) \) ≤ optimal distance . When no heuristic is used all the nodes whose distance to G is less or equal to the OD would be expanded. This tree is called Brute Force Search Tree (BFST).

\( \hat{h} \) is admissible iff: \( \forall n \rightarrow \hat{h}(n) \leq h(n) \) Eq. 1

\( \hat{h} \) is consistent for any choice of problem states n, m iff:
\( \forall n, m \rightarrow \hat{h}(n) \leq distance_{min}(n, m) + \hat{h}(m) \) Eq 2.

Informed search algorithms which use admissible heuristics are guaranteed to find the optimal solution (eventually). Consistency guarantees that when node n is expanded it has already found an optimal path to n. All admissible heuristics can be made consistent[1]. Admissible heuristics are lower bounds on the optimal distance.

The text book standard for characterizing the effect of admissible,consistent heuristics on search performance is to model the search for an optimal solution as the expansion of a Heuristic Search Tree(HST), from initial state to goal state, on which each node represents a list of successive actions taken from the initial state. HST is a sub-tree of the BFST and thus smaller[2][3]. The quality of the heuristic is defined by how small the HST is.

Among the informed search algorithms IDA* is of particular interest to us. IDA* is a linear-space version of A*. It performs a series of depth first searches, pruning a path and backtracking when the cost \( \hat{F}(n) \) of a node n on the path exceeds a \( \hat{F} \) bound C for that iteration. The initial \( \hat{F} \) bound \( C_o \) is set to the heuristic estimate of the initial state, and increases in each iteration to the lowest cost of all the nodes pruned on the last iteration, until a goal node is expanded. IDA* guarantees an optimal solution if the heuristic function is admissible.[4]

III. Problem Description
The goal stated in section I is impossible. In the best case scenario we would know how the HST expands as it grows but we would still need to know how far the HST will need to be expanded to find an optimal solution to the problem instance. Only solving the problem instance tells us the optimal distance.

The best next goal would be to predict the size of the HST given a \( \hat{F} \) bound. The main difference between IDA* and other informed search algorithms is IDA*’s iterative nature, which is \( \hat{F} \) bounded.

Each iteration of IDA* generates a \( \hat{F} \) bounded HST which is a sub-tree of the next IDA* iteration’s HST. The effect of each successive iteration is to raise the \( \hat{F} \) -bound, adding nodes to the HST’s until the final iteration. Earlier IDA* iterations can be used to predict the size of future iterations HSTs.

Existing approaches perform statistics on a problem domain by expanding HSTs for a significant number of problem instances. Individual instances are assumed to behave similarly to the average case. No approach has addressed whether it is possible to predict the HST size for future iterations with the data gathered from earlier IDA* iterations for the problem instance being solved.

Our main goal is to develop a new domain-independent model which will use the data gathered
on earlier IDA* iterations to predict the $\hat{F}$-bounded size of the HST on later iterations.

**IV. Models for predicting the size of a Search Tree**

![Example Search Trees](image)

A uniform tree is a tree on which all expanded nodes have the same branching factor and the depth of all its search paths is constant[2]. A uniform search tree is defined by its depth and branching factor. Tree A on Figure 2 is a uniform search tree.

$$N_T = \frac{B^{D+1} - 1}{B - 1}$$

Eq. 3; $N_T$=nodes created; $B$=branching factor; $D$=depth

The informed search for an optimal solution in planning is currently modeled as the expansion of a HST whose size is smaller or equal to that of a BFST. Even though neither the BFST nor the HST are uniform search trees Eq3 has been used to model the size of BFST and HSTs for significant number of problem instances across a domain[2][3]. We will use this formula as the basis of the formula to predict the BFST and HST size as we increase the $\hat{F}$-bound iterations of IDA* for individual problem instances. Its two features are B and D. Tree A has $B=2, D=3$ so $N_T=15$.

The BFST does not expand with one uniform branching factor. Each node in the BFST represent a state in the domain, and each of its children represents the result of an action applied to the parent node. Consequently each node in the BFST has a varying branching factor depending on how many actions are available from the current state. Tree B is an example of BFST.

In order to model the BFST as a uniform search trees the effective branching factor(EBF) is used in the text book model[2][3]. EBF is a simplification and represents the mean branching factor of the BFST. The BFST B has an EBF=2, D=3 so $N_T=15$.

Korf also proposed the Heuristic Branching Factor(HBF) as an alternative branching factor. HBF is the rate of growth of the HST between two IDA* iterations. It is stable in the limit of large iterations but not on the initial IDA* iterations. Korf also claims that heuristic choice does not alter the HBF[4].

If we define depth as the length of the path from I to the tree leaves then a uniform depth is not necessarily accurate for all paths of the BFST and specially not for the HST. Depending on the heuristic used some nodes will not be expanded in the HST and thus some search paths terminate earlier than the optimal depth. Also the BFST and consequently the HST may have nodes which do not expand because no action is possible from that state. Effective Depth(ED) has been suggested as an alternative to uniform depth [5]. ED is a simplification and represents the mean depth factor of the search tree.

The final HST E has a varying branching factor and a varying depth, so how do we apply Eq 3? How do we go from the quasi uniform BFST B whose size, EBF and depth we can model easily to its subtree HST E? There are infinite solutions for eq. 3 with 2 unknowns variables (EBF, ED) and only one known variable(NT).

The textbook approach would model HST E size as a uniform search tree whose EBF [2][3] has been reduced relative to the BFST EBF and whose depth is fixed to the optimal solution depth. Korf's alternative approach would be to model the HST E as a uniform tree whose EBF is the same as the corresponding BFST but whose ED has been reduced relative to the BFST[5].

Both approaches are arbitrarily fixing the depth or the branching factor as invariants. If our goal was to use Eq3 to model only the final HST E for each problem instance then it would not matter whether EBF or ED is the variant, they are equally valid.

But our goal is to use the data gathered on earlier IDA* iterations to predict the $\hat{F}$-bounded size of the HST on later iterations. HST E is the final iteration of IDA* search. The two previous
iterations generated the HSTs C & D. Using a modified version of eq 3 we can create a system of equations with 2 formulas describing C & D with two unknowns (depth and branching factor). Once depth and branching factor variables have been solved we can predict the size of the HST for any \( \hat{F} \) bound.

V. Proposed Approach & Current status

Given data gathered for earlier IDA* iterations, on a problem instance, our goal is to predict the size of the HST for future IDA* iterations up to the OD. We need a formula which models the growth of the HST up to the OD. The formula should be able to predict the size of the HST given a \( \hat{F} \) bound. We need to define the variables that, given a \( \hat{F} \) increase, can be used in the formula to predict the size of the HST.

The HST for any iteration is a subtree of the corresponding BFST. So we can describe the size of the \( \hat{F} \) bounded HST as a function of the pruning of the BFST.

\[
N_T = \frac{(EBF_{BFST} - EBF)^{\hat{F} - EDR + 1} - 1}{EBF_{BFST} - EBF - 1}
\]

Eq. 4. EBF=EBF Reduction; EDR=ED Reduction

Only two iterations are needed to numerically solve Eq. 4 for EBF and EDR. After running the first two iterations the only unknown variables are the EBF and the EDR. Once they are calculated, the size of future iterations is a function of \( \hat{F} \) only.

Solving Eq 4 with two iterations does not guarantee a perfect prediction. It assumes that both the EBF and EDR will remain constant as \( \hat{F} \) increases.

We have run thousands of instances of the Eight Puzzle for different OD with three different admissible heuristics (Out of Place, Manhattan, Relaxed Adjacency). We have come across some interesting results. If we fix EDR to 0 as the textbook model would suggest the predictions have a very low quality. If we instead fix the EBF to zero, as Korf model would suggest, the predictions are much better. But when we do not fix EBF or EDR and instead calculate them independently with 2 iterations we get the better predictions for all heuristics.

Our preliminary results support Korfs claim that EDR is a better predictor than EBF. when used as the sole variant feature mapping growth of IDA* iterations for the same problem instance, as Korf claimed. However our results support that the EBF vary depending on which heuristic we use, as in the textbook model, contrary to Korf claims. Our preliminary results validate our approach to account both for a EBF and a EDR when predicting the size of a \( \hat{F} \) bounded HST.

Domain statistical approaches like Korfs', Russell's or Nilson's are not specific to the problem instance, thus the differing effects of the heuristic can be averaged out across different problem instances. Since our approach is specific to the individual problem instance it does not have this problem.

Eq 4 depends on the EBF_{BFST} to be calculated for each iteration. On our experiments on the eight puzzle domain, EBF_{BFST} is very stable and does not change after two iterations, saving us the computational effort of expanding BFST for large depths. This is not necessarily the case for other less regular domains. More experiments are needed in other domains to test and improve our approach.

Bibliography