Predicting the Time Complexity of A*

Three Models

1. Abstract model (Pearl 84)
2. Overall distribution model (Korf, Reid and Edelekamp 2001)
3. Conditional distribution model (Zahavi et al. 2008)

Abstract Analytic model (Pearl 84)

- Assume that
  - the problem space is a tree, with no cycles
  - there is a uniform branching factor b, meaning that every node has b children
  - every edge or operator costs one unit to apply
  - there is a single goal node at depth d in the tree
- The impact of this assumption is that once we diverge from the optimal path from the root to goal, the only way to reach the goal is to backtrack until we rejoin the single optimal path.

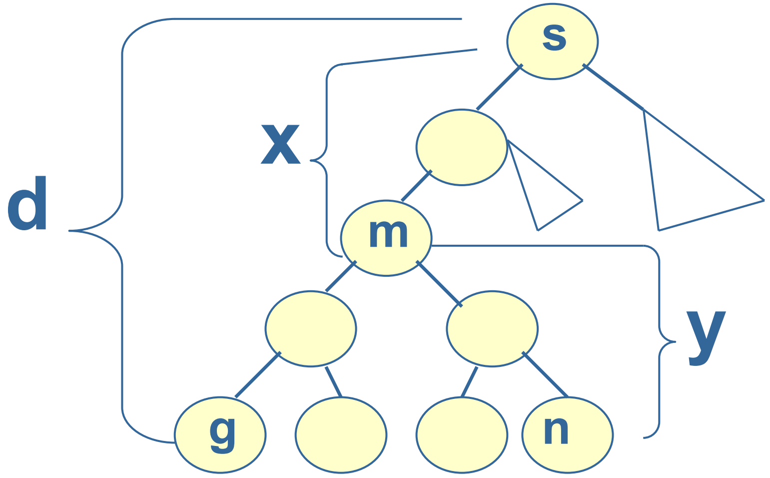
Constant Absolute Error

- We assume that the heuristic function is that it has constant absolute error. It means that *it never underestimates the optimal cost of reaching a goal by more than a constant*. Thus, for some constant k

$$\boxed{h(n) = h^*(n) - k}$$

- We need to determine how many nodes will be expanded under these assumptions.

Constant Absolute Error



- Consider a node *n* in the tree.
- Assume that the path from the start node *s* to node *n* diverges from the path to the goal at node m. Let *d* be the distance from *s* to *g* , *x* to be the distance from *s* to *m,* and *y* the distance from from *m* to *n*. Thus:
  - $h^*(n) = y + (d - x)$
  - $h(n) = y + d - x - k$

⇓

- $f(n) = g(n) + h(n) = x + y + y + d - x - k = 2y + d - k \leq d$

⇓

- $\mathbf{y \leq k/2}$

Constant Absolute Error

- For any given node m on the optimal path, the number of nodes below it, but off the optimal path, whose depth below *m* doesn't exceed k/2 is $(b-1)b^{k/2-1}$. This is because there are b -1 branches immediately below node *m* that diverge from the optimal path, the remaining child being on the optimal path and there are *b* children below every subsequent node. The number of such nodes *m* on the optimal path from which we could diverges from it is d -1, since the last node on the optimal path is the goal itself.
  - ***Thus, the total number of nodes whose total cost doesn't exceed d is $(d-1)(b-1)b^{k/2-1}$.***
- If we add the d nodes on the optimal path, this is exactly the set of nodes that are expanded by A* in the worst case. This function is O(d). (k,b are constants => $\boldsymbol{b^{k/2-1}}$ is also a constant)
  - ***Thus, the asymptotic time complexity of A* using a heuristic that has constant absolute error is linear in the solution depth.***

Constant Relative Error

- A much more realistic assumption for measurements of heuristic functions, is constant relative error. We assume that the absolute error is a ***bounded percentage of the quantity being estimated***.

$$\boxed{h(n) = \alpha\, h^*(n)}$$

- Similar derivations are possible.
- We omit them

Limitations

- There are several limitations of this model:
  - ***The abstract model makes unrealistic assumptions***. Most real problem spaces, such as Rubik's cube or the sliding tile puzzles, are graphs with cycles as opposed to trees. In such problem spaces we can reach the goal from any other state without backtracking along the path to the given state.
  - In order to determine the accuracy of the heuristic on even a single state, we need to determine the optimal solution cost to a goal from the state, which requires a great deal of computation and ***impractical for the size of problems we are trying to solve***.
- As the result of those two limitations, this model can not be used to predict the performance of A* on any real problem with a real heuristic.

Heuristic analysis on real problems [Korf, Reid and Edelekamp 2001] (KRE Model)

- Instead of characterizing a heuristic by its accuracy for purposes of its analysis, we characterize a heuristic by the distribution of heuristic values over the states in the problem space.
- We can specify this distribution by a set of parameters D(x) of states for which $h(n) \leq x$.
- We refer to this set of values as the ***overall distribution*** of the heuristic function, assuming that every state in the problem space is equally likely.
- $X \varepsilon (0, \infty)$, but for all $x \geq \max(h(n))$, $D(x) = 1$
- A way to view D(x) is that it is the probability that a state is chosen randomly and uniformly from all states in the problem space has **$h(n) \leq x$**

Characterization of the Heuristic

- ***Note***:
  - This characterization of a heuristic function in terms of its overall distribution is not a measure of the accuracy of the function.
  - The overall distribution is much easier to determine in practice than the accuracy of the heuristic function.
- The complexity of a search algorithm depends on a different distribution called the ***equilibrium distribution*** denoted as P(x).
- This is a distribution of heuristic values at a given depth of a brute-force search, in the limit of large depth.
- For the easy purpose of our course we assume that *D(x)=P(x)=probability that a node n has h(n)≤x*

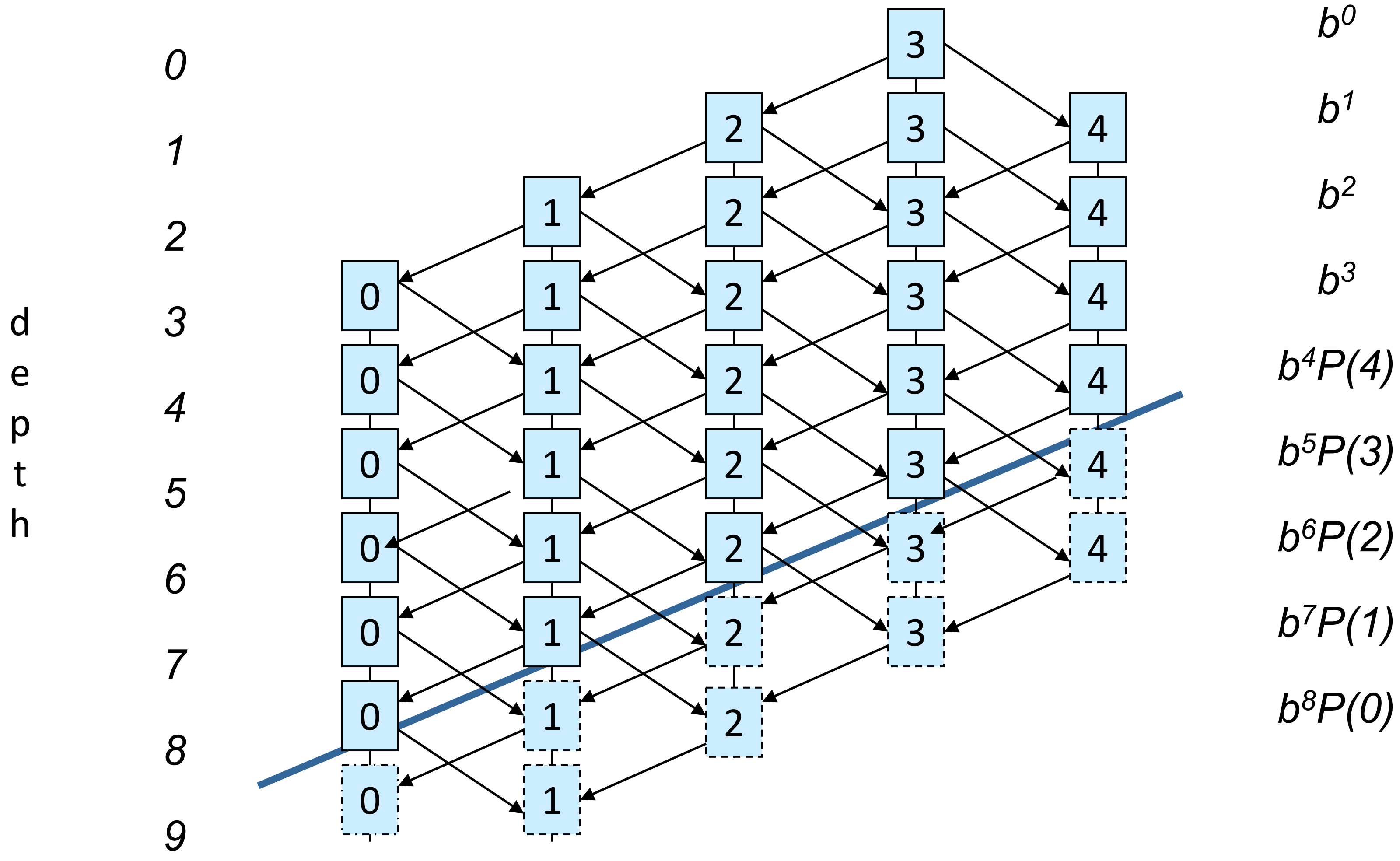
Assumptions of Analysis

- Our search algorithm doesn’t check for or detect state that have been previously generated. While this is not true for A*. It is true for the linear space variants. Thus, the multiple nodes that correspond to the same state of the problem are counted separately.
- All edges have unit cost, and hence the cost of a solution is the number of edges in the solution path.
- The heuristic function is also integer valued.
- The heuristic is consistent.

Task

- Our task is to determine n(b,d,P) the asymptotic worst-case number of nodes generated by A* in searching a problem space with branching factor b, optimal solution depth d, and a heuristic is characterized by the equilibrium distribution P(x).

An Example Search Tree



An Example Search Tree

- On the figure:
  - **the vertical axis** represents the depth of node below the start node.
  - **the horizontal axis** represents the heuristic value.
  - **each box** represents a set of nodes at the same depth with the same heuristic value, indicated by the number in the box.
  - **the arrows** represent the relationship between parent and child nodes. h(child) $\geq$h(parent) -1.
  - **the solid boxes** represent "fertile" nodes which will be expanded.
  - **the dotted boxes** represent "sterile" nodes that will not be expanded, because their total cost exceeds the optimal solution cost.
  - **the thick diagonal line** separates the fertile nodes from the sterile nodes.
- In this particular example, the maximum value of the heuristic is 4, and the depth of solution d is 8 moves. We chose 3 for the heuristic value of the initial state.

Nodes Expanded as a Function of Depth

- We will count the number of node expansions, which is the number of fertile nodes in the graph. The number of nodes generated is simply *b* times the number expanded.
- There is a single state at depth 0. This root node generates b children, whose heuristic value range from 2 to 4, inclusive. Each of these nodes generate b nodes, whose heuristic value will range from 1 to 4, giving a total of b^2 nodes at depth 2. Since the goal is assumed to be at depth 8, in the worst case, all nodes n whose total cost $f(n) = g(n) + h(n) \leq 8$ will be expanded. Since 4 is a maximum heuristic value, all nodes down to depth 8 - 4 = 4 will be expanded, as in brute force search. Down to this depth, the number of nodes at depth d will be b^d. Note that $P(4) = 1$, and hence $b^4P(4) = b^4$.
- In general, down to depth d - m, where m is max heuristic value, all nodes are expanded.

Nodes Expanded as a Function of Depth

- The nodes expanded at depth 5 are fertile nodes, which are those with $f(n) = g(n) + h(n) = 5 + h(n) \leq 8 \quad \Rightarrow h(n) \leq 3$. In equilibrium distribution, the fraction of nodes at depth 5 with $h(n) \leq 3$ is $P(3)$. Since all nodes at depth 4 are expanded, the total number of nodes at depth 5 is b^5, and the number of fertile nodes at depth 5 is $b^5P(3)$.

- While there are nodes at depth 6 with all possible heuristic values, their distribution in no longer equal to the equilibrium distribution. The number of nodes at depth 6 with $h(n) \leq 2$ is completely unaffected by this pruning and is the same as it would be in a brute force search at depth 6, or $b^6P(2)$.

Nodes Expanded as a Function of Depth

- Due to consistency of the heuristic function, all the possible parents of fertile nodes are themselves fertile. ***Thus, the number of nodes with each heuristic value to the left of the diagonal line is the same as it would be in a brute force search to the same depth.***
- If the heuristic were inconsistent, then the distribution of fertile nodes would change at every level where pruning occurred, making analysis much more complex.
- In general, the number of fertile nodes at depth *i* is $b^i P(d - i)$

$$\mathbf{N(b,d,P) = \sum_{i=0}^{d} b^i P(d-i)}$$

Problems with KRE

- Did not work in practice. It is only true for a large sample of start states but not for a single state.
  - states do not behave according to P unless they are very far away from the start state.

- Does not work for inconsistent heuristics.

KRE for Inconsistent Heuristics

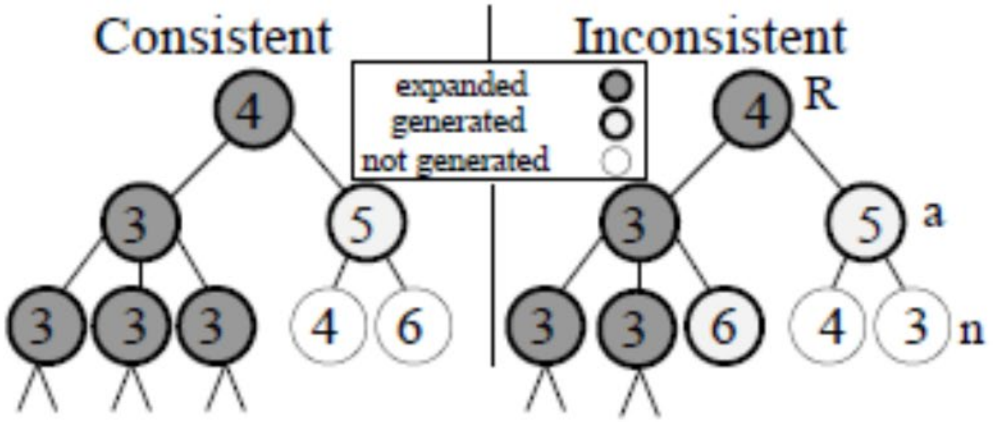


Figure 1: Consistent versus inconsistent heuristics

- KRE counts the fertile nodes (those above the horizontal line)
- Not valid for inconsistent heuristics
- Fertile nodes may not be generated

CDP: Conditional Distribution Prediction (Zahavi et al. 2008)

- Instead of p(v) – static distribution, we store p(v,vp) the probability that a value of a node with state s is h(s)=v given that its parent node with state p value was h(p)=vp.

h(s) \ h(p)	•••	6	7	8	9	10
•••						
6		0.30	0.11	0.00	0.00	0.00
7		0.60	0.45	0.21	0.00	0.00
8		0.00	0.44	0.70	0.67	0.00
9		0.00	0.00	0.09	0.32	0.89
10		0.00	0.00	0.00	0.01	0.11

(a) Consistent heuristic

h(s) \ h(p)	•••	6	7	8	9	10
•••						
6		0.17	0.11	0.06	0.03	0.02
7		0.36	0.38	0.33	0.25	0.19
8		0.37	0.44	0.53	0.60	0.62
9		0.02	0.03	0.05	0.08	0.14
10		0.00	0.00	0.00	0.01	0.01

(b) Inconsistent heuristic

Figure 2: A portion of the p_2 matrix for Rubik's Cube

CDP

$$\approx \tilde{N}_i(s,d,v) = \sum_{v_p=0}^{d-(i-1)} \tilde{N}_{i-1}(s,d,v_p) \cdot b_{v_p} \cdot p(v|v_p)$$

- $N_i(s,v)$= the number of nodes expanded by IDA* at depth i from s with a heuristic of v

- N_0=1 for v=h(start) and 0 otherwise.

$$\mathrm{CDP}_1(s,d) = \sum_{i=0}^{d} \sum_{v=0}^{d-i} \tilde{N}_i(s,d,v)$$

Experimental Results for Rubik's Cube

		KRE		CDP_1		CDP_2	
h	IDA*	Prediction	Ratio	Prediction	Ratio	Prediction	Ratio
5	30,363,829	8,161,064	0.27	48,972,619	1.61	20,771,895	0.68
6	18,533,503	8,161,064	0.44	17,300,476	0.93	13,525,425	0.73
7	10,065,838	8,161,064	0.81	7,918,821	0.79	9,131,303	0.91
8	6,002,025	8,161,064	1.36	5,094,018	0.85	6,743,686	1.12
9	3,538,964	8,161,064	2.31	3,946,146	1.12	5,240,425	1.48

Results for different start state heuristic values (h) for a consistent heuristic with an IDA* threshold of d = 12

		KRE		CDP_1		CDP_2	
d	IDA*	Prediction	Ratio	Prediction	Ratio	Prediction	Ratio
Dual							
8	36	257	7.14	31	0.86	36	1.00
9	518	3,431	6.62	418	0.81	508	0.98
10	6,809	45,801	6.73	5,556	0.82	6,792	1.00
11	92,094	611,385	6.64	74,037	0.80	90,664	0.98
12	1,225,538	8,161,064	6.66	987,666	0.81	1,210,225	0.99
13	16,333,931	108,937,712	6.67	13,180,960	0.81	16,154,640	0.99
Random Symmetry							

With an inconsistent heuristic