Single Agent Search

Lecture 8
IDA* Performance

UNIVERSITY of DENVER
DANIEL FELIX RITCHIE SCHOOL OF ENGINEERING & COMPUTER SCIENCE

State Space Analysis

- What is “*b*”
  - Used the value, but haven’t analyzed where it comes from
  - Analyze problems to compute *b*
  - Predict performance of IDA* (node exp)

Easy Example

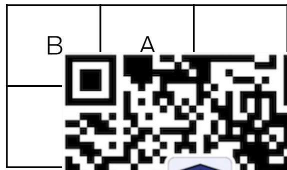
- 2x2 sliding tile puzzle (3-puzzle)
- Branching factor?
  - 2 if we include parents in analysis
  - 1 if we don’t

Tougher Example

- 3x2 5-puzzle
  - 2/6 states are edge states
    - branching factor is 3 (2 w/o parent)
  - 4/6 states are corner states
    - branching factor is 2 (1 w/o parent)
  - b = 2/6*3 + 4/6*2 = 2.3333
    - Assumes we are equally likely to be any state

Time spent in each state

- Label states with types A/B
- Without pruning parents
  - A → 2B + 1A
  - B → 1A + 1B
- Use this to determine branching factor



Asymptotic branching factor

- Consider a tree from any particular start state
  - Compute the number of nodes at each level
  - *b* is the ratio of nodes at level *n* to *n*+1
    - (As n gets large)
- (Demonstrate example on 5-puzzle)

Asymptotic Branching Factor

- At any particular level of the tree how many states of type A and of type B are there?
  - Eventually the fraction of A and B states should stabilize
  - The number of A nodes at the next level will be the same as the previous level
  - The actual count will differ by a multiple of *b*

Asymptotic branching factor

- So, we know:
  - $b f_a = f_a + f_b$
  - $b f_b = 2f_a + f_b$
  - $f_a + f_b = 1$
- Solve for *b* and get $b = 1+\sqrt{2}$

UNIVERSITY of DENVER
DANIEL FELIX RITCHIE SCHOOL OF ENGINEERING & COMPUTER SCIENCE

Removing reverse operators

- Now we need 4 variables
  - A - center (from a center)
  - B - center (from a corner)
  - C - corner (from a center)
  - D - corner (from a corner)

UNIVERSITY of DENVER
DANIEL FELIX RITCHIE SCHOOL OF ENGINEERING & COMPUTER SCIENCE

Removing reverse operators

- A → 2C
- B → A + C
- C → D
- D → B

- $b f_a = f_b$
- $b f_b = f_d$
- $b f_c = 2f_a + f_b$
- $b f_d = f_c$
- $f_a + f_b + f_c + f_d = 1$

$$b^4 - b - 2 = 0$$

- Solve for b = 1.35

UNIVERSITY of DENVER
DANIEL FELIX RITCHIE SCHOOL OF ENGINEERING & COMPUTER SCIENCE

Rubik's Cube

- Naive approach:
  - 18 moves (90°, -90°, 180° turns per face)
- Better approach:
  - 15 moves (disallow previous face)
- Even better approach:
  - Restrict successive turns of face, opposite face and then first face

UNIVERSITY of DENVER
DANIEL FELIX RITCHIE SCHOOL OF ENGINEERING & COMPUTER SCIENCE

Rubik's Cube - Best approach

- Label 3 faces "first faces"
  - Faces should all border each other
- Other faces are "second faces"
- Disallow turning a first face after turning its associated second face
  - b = 13.3487

Rubik's Cube

- Recurrence
  - F → 6F+9S
  - S → 6F+6S
- Equations
  - bF = 6S+6F
  - bS = 9F+6S
  - S+F=1
- F = 0.4494897428
- S = 0.5505102572

Predicting performance of IDA*

- Want to predict the performance of IDA* in general
- What is the asymptotic effect of having a heuristic?
  - Pearl, 84 predicted:
    - Search size b^d
    - With better heuristic a^d with $a < b$
- Analysis isn't necessarily incorrect; depends on assumptions

Predicting IDA* performance

- What do we need to determine performance?
  - Assume consistent heuristic
    - Analysis has been performed for inconsistent heuristics, but more complicated
  - Assume cost threshold of *c* on last iteration
    - All nodes with f(n) < c must be expanded
    - All nodes with f(n) ≤ c may be expanded
  - Information about the heuristic

Characterizing a heuristic - initial assessment

- Suppose there are N states in the world
  - How many states have h(a) = a for *a* in 1…N?
    - Define this as d(a)
  - D(h) is: (∑d(a)) / N for a = 1…h
    - The percentage (fraction) of states with h(s) ≤ h
- **Heuristic distribution**

Characterizing heuristic & search space

- Heuristic distribution may not represent how we exactly encounter states in our sea
- This is controlled by the **equilibriu**
  - What is the actual distribution of in practice
    - A bit strange, because we sea start state
    - Cleaned up in later analysis

Equilibrium distribution

- Consider 5-puzzle, blank is:
  - 35.321% of time in side position
  - 64.679% of time in corner position
- Consider the heuristic distribution according to where the blank is

Equilibrium Distribution

- P(h)
  - The probability of a node having a heuristic ≤ h
  - $P(h(n) < h \mid side) \cdot P(side) +$
    $P(h(n) < h \mid corner) \cdot P(corner)$
- Example table with manhattan distance (next slide)
  - P(h) is different from D(h)

Table 2
Heuristic distributions for Manhattan distance on the Five Puzzle

h	States	Sum	*D*(*h*)	Corner	Side	Csum	Ssum	*P*(*h*)
0	1	1	0.002778	1	0	1	0	0.002695
1	2	3	0.008333	1	1	2	1	0.008333
2	3	6	0.016667	1	2	3	3	0.016915
3	6	12	0.033333	5	1	8	4	0.033333
4	30	42	0.116667	25	5	33	9	0.115424
5	58	100	0.277778	38	20	71	29	0.276701
6	61	161	0.447222	38	23	109	52	0.446808
7	58	219	0.608333	41	17	150	69	0.607340
8	60	279	0.775000	44	16	194	85	0.773012
9	48	327	0.908333	31	17	225	102	0.906594
10	24	351	0.975000	11	13	236	115	0.974503
11	8	359	0.997222	4	4	240	119	0.997057
12	1	360	1.000000	0	1	240	120	1.000000

University of Denver — Daniel Felix Ritchie School of Engineering & Computer Science

What does a tree look like?

- Given a node & consistent heuristic:
  - h-value is either 1 greater than o[riginal] or original heuristic value
- Divide children into buckets
  - Sample tree
    - h = 0…3, c = 5

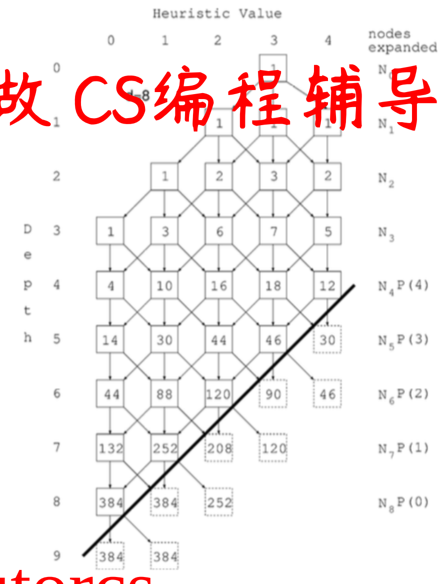


Fig. 4. Sample tree for analysis of IDA*.

Search depth 8

Max heuristic = 4

University of Denver — Daniel Felix Ritchie School of Engineering & Computer Science

What is the total work?

- Expected number of nodes expanded

$$E(N,c,P) = \sum_{i=0}^{c} N_i P(c-i)$$

  - N_i is simply b^i where b is the asymptotic branching factor
- *Using this, what is the heuristic branching factor?*

University of Denver — Daniel Felix Ritchie School of Engineering & Computer Science

Analysis

- Heuristic branching factor:

$$\frac{E(N,c,P)}{E(N,c-1,P)} = \frac{\sum_{i=0}^{c} N_i P(c-i)}{\sum_{i=0}^{c-1} N_i P(c-i-i)}$$

$$= \frac{b^0 P(c) + b^1 P(c-1) + b^2 P(c-2) + \cdots + b^c P(0)}{b^0 P(c-1) + b^1 P(c-2) + b^2 P(c-3) + \cdots + b^{c-1} P(0)}$$

$$\approx b$$

Analysis

- In an exponential domain the effect of a heuristic is to keep the branching factor the sa
- If we increase the heuristic, we dec at which we get cutoffs
  - In exponential spaces, improving just decreases the effective level

Discussion

- Perfectly predicts nodes expanded for 8-puzzle if:
  - Average over all starting states
  - Search to fixed depth *c* (past the goal)
- Not necessarily a good predictor for an actual problem