Heuristics used for A\* Search

A combination of the following heuristics were used to determine which successor would be chosen

1. Decrement by one if the blocks in the stack other than the first are in the reverse order i. e value[ i ]=value [ i+1 ]+1.

This way it’s easier to put them the right way onto the first stack

1. Decrement by one if the a block in the first stack sits on top of a block with a value lesser than one. Ex: C on B
2. Increment by one if the blocks in the first stack don’t correspond to their positions.
3. Increment by one for every other block out of place.

These heuristics are combined with g(n) i.e f(n)=g(n)+h(n)

Observations:

stacks are represented by [ bottom...top ]

|  |  |  |  |
| --- | --- | --- | --- |
|  | No. of goal tests(time complexity) | Max frontier(space complexity) | Path length |
| [D,E,B,A,C]  [ ]  [ ] | 105 | 110 | 11 |
| [ ]  [C]  [F,B,A]  [E,D ] | 15 | 64 | 6 |
| [F,E]  [ ]  [C,G,D]  [ ]  [B]  [A] | 478 | 7175 | 14 |
| [A]  [C,B,E]  [D] | 18 | 28 | 6 |
| [E,C,G,B,F,D]  [ ]  [ ]  [A] | 796 | 2811 | 15 |

It is seen that the number of iterations increases with more number of blocks and fewer stacks. But configuration of blocks also plays a factor in the determination of these values.