Heuristic: Modified Blocks-Out-Of-Place Heuristic (Block Positioning)

This heuristic is an extension of the H1 Heuristic “blocks-out-of-place” by evaluating the block positioning more comprehensively. Heuristic H2 is designed to provide a more detailed evaluation of the current state in comparison to the goal state. It goes beyond the simple count of blocks and considers the precise positioning of blocks in relation to one another in both states. This heuristic focuses on the arrangement of blocks within the stacks and checks if they meet specific criteria based on the goal state. The heuristic starts by iterating over each block in the goal state, considering each stack in the goal state individually. For each block in the goal state, Heuristic H2 tries to find the corresponding block in the current state. It does this by searching through the stacks in the current state to locate the block with the same label. For each block in the goal state, Heuristic H2 evaluates its positioning with respect to neighboring blocks (if they exist) in both the current state and the goal state. It considers the following scenarios:

a. Block on Top: If a block A in the goal state is supposed to be on top of another block B, Heuristic H2 checks if in the current state, block A is indeed on top of B. If not, it adds a penalty of 2 to the heuristic.

b. Block Underneath: If a block A in the goal state is supposed to be underneath another block C, Heuristic H2 checks if in the current state, block A is indeed underneath C. If not, it adds a penalty of 2 to the heuristic.

c. Block on Top or Underneath: Even if a block A in the goal state is not precisely on top of B or underneath C, Heuristic H2 provides a partial credit. If block A is either on top of B or underneath C in the current state, it adds a penalty of 1 to the heuristic.

The heuristic score is accumulated as the sum of penalties for all evaluated blocks in all stacks in the current state, and the final heuristic value (h2(n)) represents the total accumulated penalty score, representing how far the current state differs from the goal state in terms of block positioning.

Examples:

Initial State:

A B C

D E F

G H I

Goal State

A B C

G E F

D H I

In this example, block A in the current state is not on top of block G as it should be in the goal state. Additionally, block D is not underneath block G in the current state. Therefore, Heuristic H2 (h2(n)) assigns a penalty of 4 to this state, 2 for each infraction: a penalty of 2 for block A not being on top of G, and a penalty of 2 for block D not being underneath G.

Initial State:

A B C

D E F

G H I

Goal State

A B C

D E F

H G I

In this example, Block H in the current state is not on top of block G as it should be in the goal state. However, block D is correctly underneath block H. Therefore, Heuristic H2 (h2(n)) assigns a penalty of 2 to this state: a penalty of 2 for block H not being on top of G.

The performance of this heuristic is measured in the following table:

| Test Case | Solution Length | Num Iterations | Max Queue Size |
| --- | --- | --- | --- |
| A03 | 3 | 8 | 20 |
| A04 | 4 | 35 | 58 |
| A05 | 5 | 85 | 149 |
| A06 | 6 | 182 | 245 |
| A07 | 7 | 346 | 469 |
| A08 | 8 | 729 | 863 |
| A09 | 9 | 3009 | 2280 |
| A10 | 10 | 2459 | 1962 |
| A11 | 11 | 3506 | 2203 |
| B03 | 3 | 48 | 551 |
| B04 | 4 | 46 | 614 |
| B05 | 5 | 192 | 2347 |
| B06 | 6 | 376 | 3303 |
| B07 | 7 | 37770 | 298163 |
| B08 | 8 | 8161 | 64974 |
| B09 | 8 | 91173 | 666964 |
| B10 | 9 | 133162 | 940806 |
| B11 | 9 | 265572 | 1769797 |
| B12 | 9 | 679179 | 4503086 |
| B13 | FAILED | 1000000 | 5481842 |
| B14 | FAILED | 1000000 | 6013336 |
| B15 | FAILED | 1000000 | 5823899 |
| B16 | FAILED | 1000000 | 5407642 |
| B17 | FAILED | 1000000 | 6134162 |
| B18 | FAILED | 1000000 | 6141714 |
| B19 | FAILED | 1000000 | 6243834 |
| B20 | FAILED | 1000000 | 6082937 |