Consistency and Admissibility Proof(kinda) for Total Manhattan Distance Heuristic for Cube problem

Wang Xiyu

```cpp
#include <bits/stdc++.h>
using namespace std;
int r, c;
double heuristic(const vector<vector<int>>& curr, const vector<vector<int>>& goal) {
    vector<int> misplaced_idx(r, -1); // to keep track of last misplaced element, so to not
        using it again
    double cost = 0;
    for (int i = 0; i < r; i++) {
        misplaced_idx[i] = -1;
    }

    for (int i = 0; i < r; i++) {
        for (int j = 0; j < c; j++) {
            int g = goal[i][j]; // what it should be
            if (curr[i][j] == g) {
                continue; // not misplaced, cont;
            } else {
                int n = 0;
                for (int p = 0; p < r; p++) {
                    for (int q = 0; q < c; q++) {
                        if (curr[p][q] == g // find a tile that should be placed here
                                && (r * p + q > misplaced_idx[g]) // its index should be larger
                                    than the index of the last tile of the same type being used
                                    to replace misplaced tile
                                && curr[p][q] != goal[p][q]) { // this tile will be moved thus it
                                    shoulnt be at the place it should be as well
                            n = r * p + q; // if find one, update the index of the tile to
                                replace the misplaced tile
                            misplaced_idx[g] = n;
                        }
                    }
                }
                cost += min(abs(n % c - j), abs(j - n % c)); // min: can rotate in 2 direction,
                    take the shorted way
                cost += min(abs(n / c - i), abs(i - n / c)); // abs: absolute distance in hor
                    and vert dir
                                                    // n % c - j: hor manhatten dist
                                                    // n / r - i: vert manhatten dist
            }
        }
    }
    return cost / (r * c);
}
```

In worst case scenario each move either horizontal or vertical will increment the total manhattan distance by $\max(r, c)$, when all original tiles are in-place and the move shifts them away by 1 each.

$$\forall n, n' \in State, |h(n) - h(n')| \leq \frac{\max(r, c)}{x}$$

to make this maximum possible change lower than the actual cost of this step, let $x = rc$.

$$\forall n, n' \in State, |h(n) - h(n')| \leq \min(\frac{1}{c}, \frac{1}{r})$$

We also have

$$\forall n, n' \in State, a \in Action, c(n, a, n') = 1$$

$$h(n) \leq \min(\frac{1}{c}, \frac{1}{r}) + h(n') \leq c(n, a, n') + h(n') \implies \text{h(n) is consistent} \implies \text{h(n) is admissible} \quad \square$$

Question:

If I am on the right track, why i cannot use $x = max(r, c)$ as per my initial attempt, since it also bounds $\Delta \sum$ Manhattan distance ≤ 1