Wykorzystane Heurystyki:

h1(Hamming) - Prosta heurystyka zwracająca ilość liczb nie na swoim miejscu.

h2(Manhattan) - Heurystyka zwracająca sumę odległości(liczba rzędów i wierszy różnicy) liczb od ich docelowych pozycji

Generowanie permutacji:

Wykorzystuję 2 sposoby generowania przestawionych tablic

generateSolvablePuzzle - wykonuje (n*n)-2 przestawień losowych liczb w docelowej tablicy i sprawdza czy da się taką układankę rozwiązać, jeśli nie to losuje aż do skutku

makeRandomMoves - wykonuje k losowych ruchów zaczynając w pozycji docelowej tablicy. Zawsze da się rozwiązać, bo wystarczy cofnąć te ruchy.

Opis działania:

1. Inicjalizacja

Ustawiany jest licznik odwiedzonych stanów.

Tworzony jest zbiór możliwych stanów do odwiedzenia (posortowanych według sumy kosztu dotarcia i heurystyki).

Tworzona jest mapa kosztów dotarcia do danego stanu (g).

Tworzona jest mapa kosztów całkowitych (f = g + h).

Tworzona jest mapa previous, która śledzi, z jakiego stanu przeszliśmy do danego.

Tworzony jest zbiór stanów odwiedzonych.

Stan początkowy zostaje dodany do otwartej listy (możliwych stanów).

2. Główna pętla (dopóki są stany do sprawdzenia)

a. Wybór najlepszego stanu

Ze zbioru możliwych stanów wybierany jest ten z najmniejszym kosztem całkowitym f.

Jeśli to jest stan końcowy (czyli rozwiązanie), to przerywamy pętlę i odtwarzamy drogę do celu.

b. Oznaczenie stanu jako odwiedzonego

Usuwamy ten stan z listy otwartej.

Dodajemy go do zbioru odwiedzonych.

3. Sprawdzenie sąsiadów aktualnego stanu

Dla każdego możliwego ruchu (czyli przesunięcia pustego pola w górę, dół, lewo lub prawo), tworzony jest nowy stan (sąsiad):

a. Pomijanie już odwiedzonych stanów

Jeśli sąsiad został już odwiedzony, jest pomijany.

b. Obliczanie kosztu dotarcia do sąsiada

Jeśli jeszcze nie mieliśmy informacji o koszcie dotarcia, obliczamy go i aktualizujemy mapy kosztów.

c. Porównanie nowej ścieżki do już znanej

Jeśli dotarcie przez bieżący stan do sąsiada jest lepsze (czyli koszt jest mniejszy lub równy), to:

Aktualizujemy ścieżkę (previous),

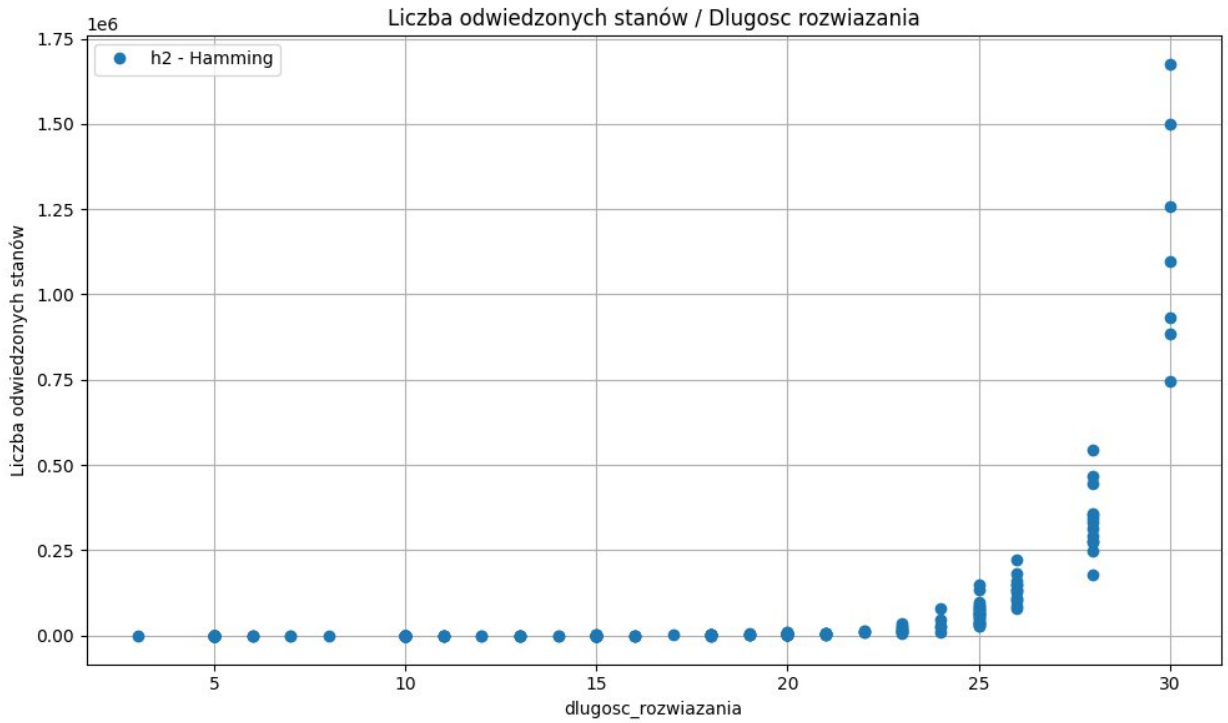
Aktualizujemy koszt dojścia (g),

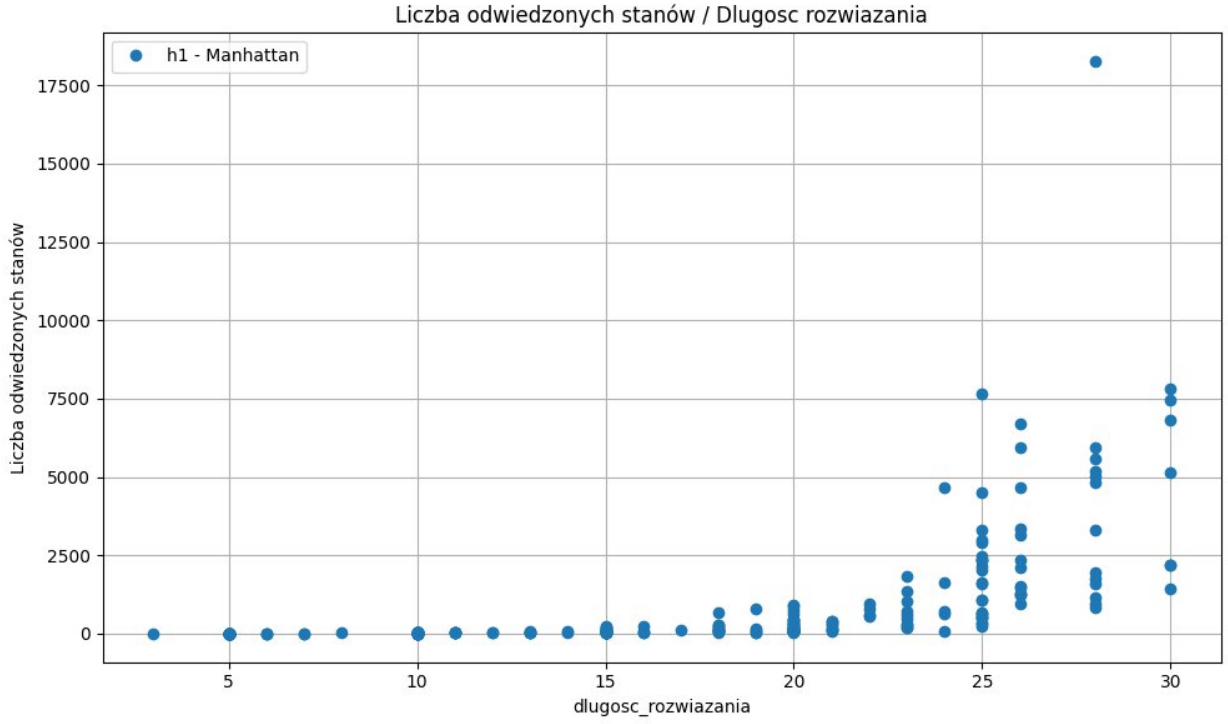
Obliczamy nowy koszt całkowity (f),

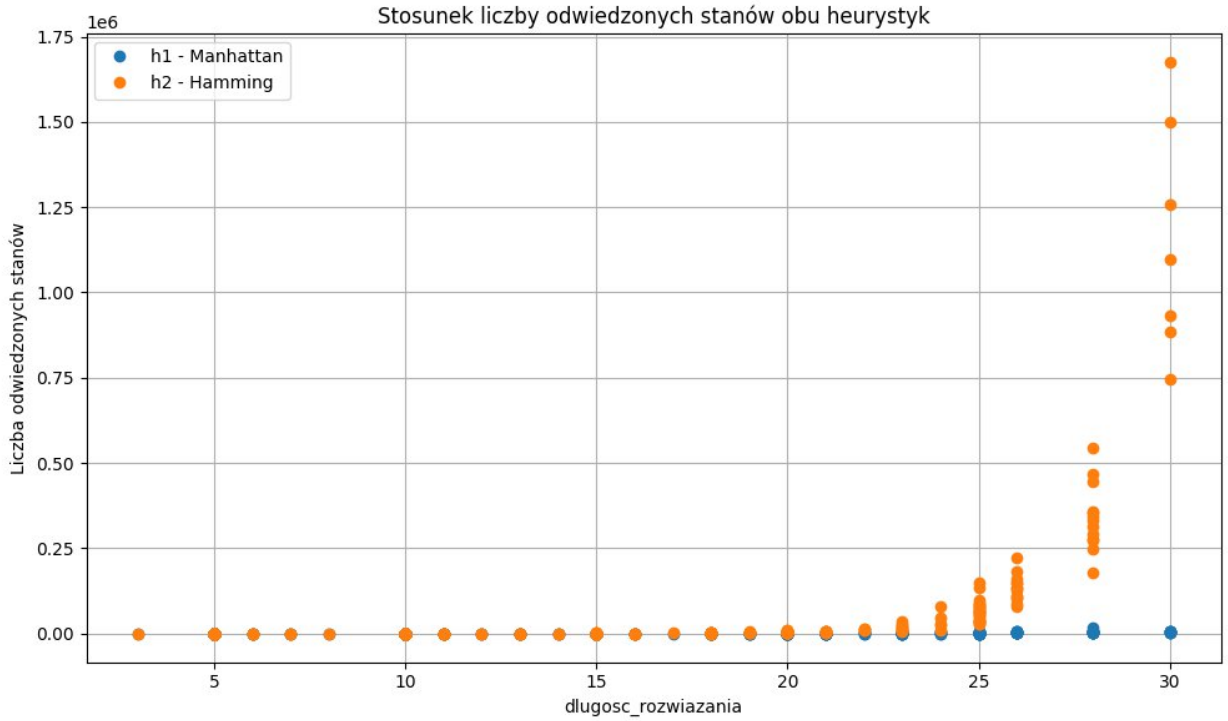
Dodajemy stan do listy możliwych.

Wnioski:

Heurystyka h1 jest o wiele wydajniesza rozwiązując zadania stanowczo szybciej i odwiedzając mniejszą liczbę stanów.


Liczba odwiedzonych stanów / Dlugosc rozwiazania
1e6
h2 - Hamming
Liczba odwiedzonych stanów
dlugosc_rozwiazania


Liczba odwiedzonych stanów / Dlugosc rozwiazania
h1 - Manhattan
Liczba odwiedzonych stanów
dlugosc_rozwiazania


Stosunek liczby odwiedzonych stanów obu heurystyk
1e6
h1 - Manhattan
h2 - Hamming
Liczba odwiedzonych stanów
dlugosc_rozwiazania
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
5
10
15
20
25
30