Informatyka Stosowana, studia dzienne, inż I st. semestr IV

Sztuczna inteligencja i systemy ekspertowe 2021/2022

Prowadzący: Krzysztof Lichy poniedziałek, 13:45

Data oddania: _____ Ocena: _____

Paulina Sidor 229995
Dorota Wiechno 236690

Zadanie 1: Piętnastka

1. Cel

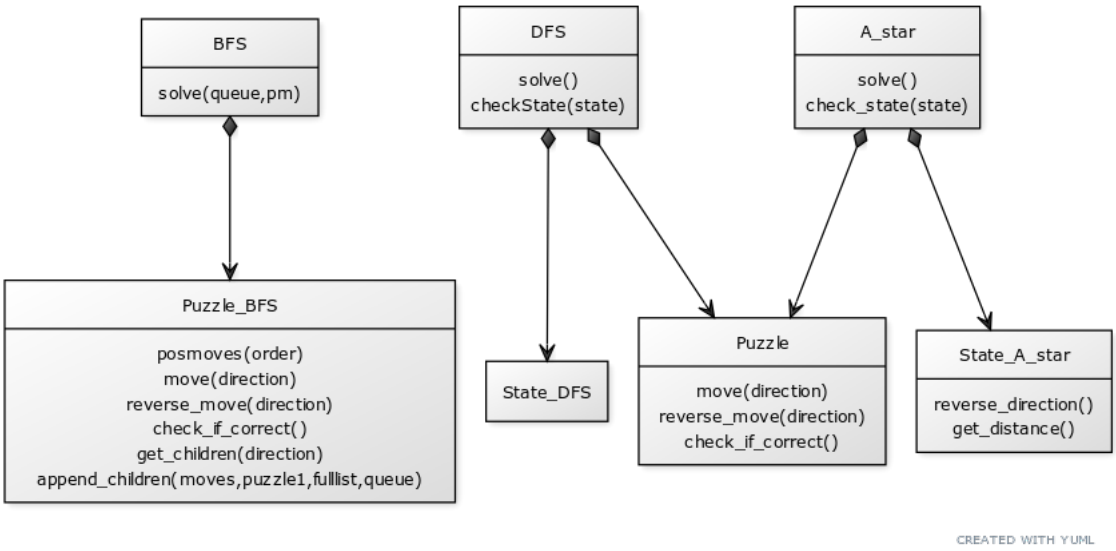
Napisanie programu, który będzie rozwiązywał układankę przy użyciu różnych metod przeszukiwania przestrzeni stanów - strategii "wszerz", strategii "w głąb" oraz strategii A*. Przebadanie, jak powyższe metody przeszukiwania przestrzeni stanów zachowują się przy rozwiązywaniu układanek .

2. Wprowadzenie

Piętnastka to gra logiczna zbudowana z pudełka, w którym znajduje się 15 kwadratowych klocków o jednakowych rozmiarach ułożonych w kwadrat 4×4 i ponumerowanych od 1 do 15. Jedno miejsce jest puste i umożliwia przesuwanie sąsiednich klocków względem siebie. Celem gry jest ułożenie klocków odwzorowując stan poniżej.

Algorytmy odnajdujące rozwiązanie Piętnastki polegają na przeszukaniu drzewa stanów. Poszukujemy stanu terminalnego jakim jest rozwiązana układanka. DFS i BFS to strategie ślepe. Polegają na założeniu, że zanim osiągniemy stan terminalny, oszacowanie kosztów ścieżki jest niemożliwe. Algorytm BFS przeszukuje po kolei poszczególne warstwy. Natomiast DFS to przeszukiwanie w głąb. Aby przetworzyć stan badamy wszystkich jego potomków.
Algorytm A* jest przykładem strategii heurystycznej. Korzystamy z kolekcji stanów Q. W kolejnych krokach wybieramy z Q stan o minimalnej wartości C(s)+h(s) obliczanych zgodnie z wybraną heurystyką. Dodajemy do Q potomków tego stanu. Wykonujemy aż do osiągnięcia stanu terminalnego.
Metryka Hamminga - liczba elementów nie na swoim miejscu
Metryka Manhattan - suma ruchów potrzebnych do osiągniecia swojego miejsca dla każdego elementu [1]

3. Opis implementacji

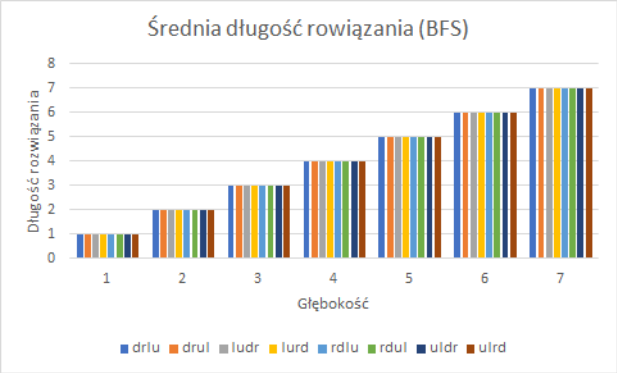


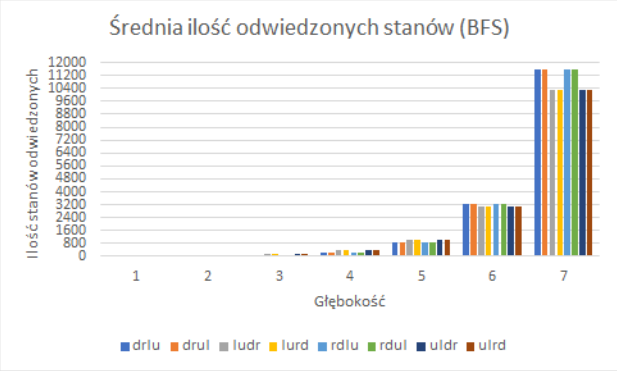
Aplikacja została napisana w języku Python. Zaprojektowano klasy odpowiadające poszczególnym algorytmom (BFS, DFS i Astar) oraz klasy pomocnicze (dwie klasy Puzzle i dwie klasy State) Algorytmy DFS i Astar korzystają z tej samej klasy Puzzle.

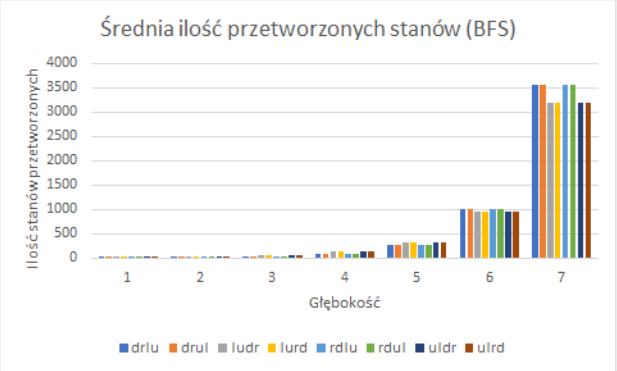
4. Materiały i metody

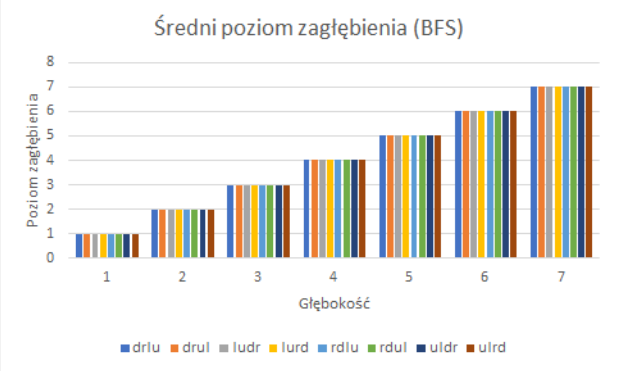
Do przeprowadzenia badań wykorzystano skrypty i narzędzia generujące układanki oraz wywołujące rozwiązywanie układanek z kolejnymi parametrami. Wygenerowano 413 układów, różniących się od rozwiązanego układu ilością ruchów od 1 do 7. Rozwiązano układy za pomocą trzech algorytmów. W przypadku DFS i BFS sprawdzono 8 różnych kolejności wyboru stanów. W przypakdu A* rozwiązano układanki z obiema heurystykami. (Hamming i Manhattan) Uzyskane wyniki przedstawiono w postaci wykresów.

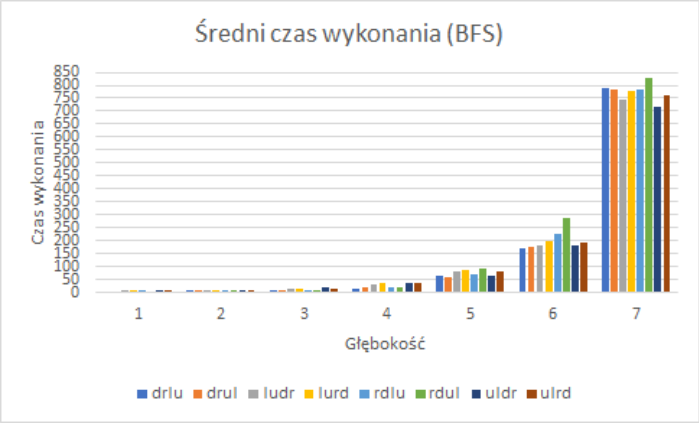
5. Wyniki

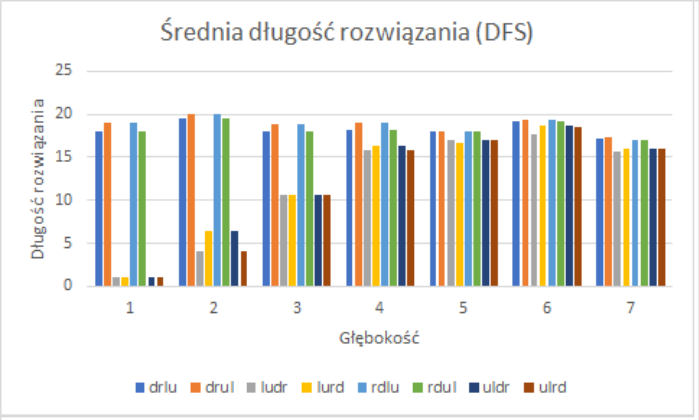

Średnia długość rowiązania (BFS)
Długość rozwiązania
0
1
2
3
4
5
6
7
8
1
2
3
4
5
6
7
Głębokość
drlu drul ludr lurd rdlu rdul uldr ulrd

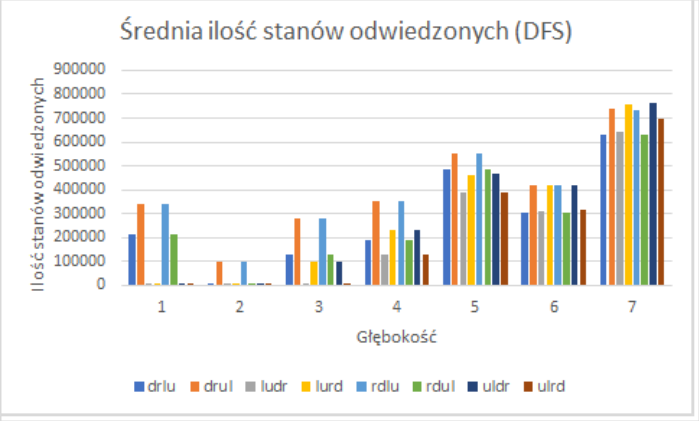

Średnia ilość odwiedzonych stanów (BFS)
Ilość stanów odwiedzonych
0
800
1600
2400
3200
4000
4800
5600
6400
7200
8000
8800
9600
10400
11200
12000
1
2
3
4
5
6
7
Głębokość
drlu drul ludr lurd rdlu rdul uldr ulrd

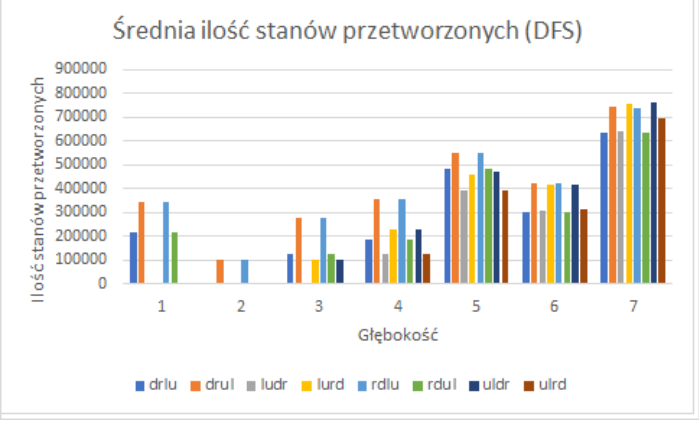

Średnia ilość przetworzonych stanów (BFS)
Ilość stanów przetworzonych
0
500
1000
1500
2000
2500
3000
3500
4000
1
2
3
4
5
6
7
Głębokość
drlu drul ludr lurd rdlu rdul uldr ulrd

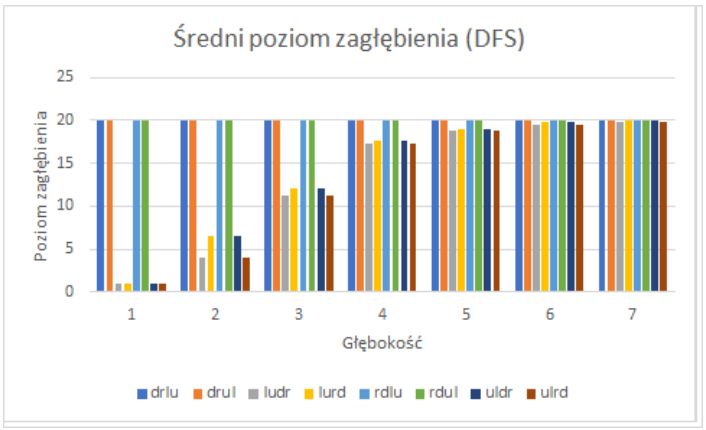

Średni poziom zagłębienia (BFS)
Poziom zagłębienia
0
1
2
3
4
5
6
7
8
1
2
3
4
5
6
7
Głębokość
drlu drul ludr lurd rdlu rdul uldr ulrd

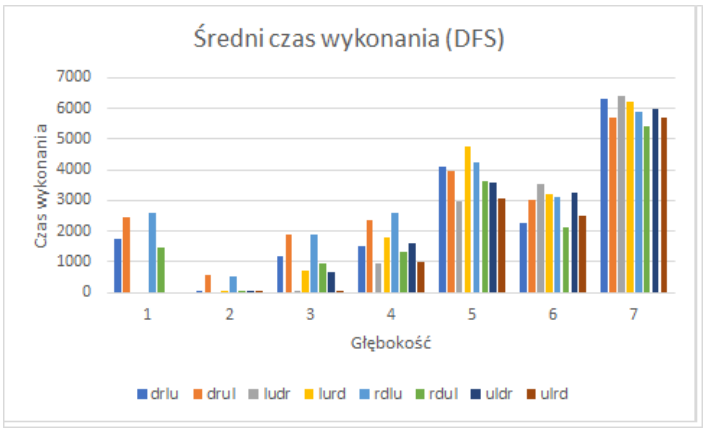

Średni czas wykonania (BFS)
Czas wykonania
850
800
750
700
650
600
550
500
450
400
350
300
250
200
150
100
50
0
1
2
3
4
5
6
7
Głębokość
drlu
drul
ludr
lurd
rdlu
rdul
uldr
ulrd

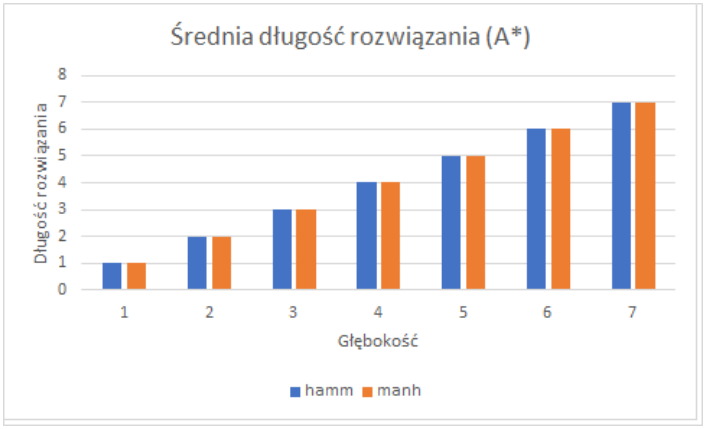

Średnia długość rozwiązania (DFS)
Długość rozwiązania
25
20
15
10
5
0
1
2
3
4
5
6
7
Głębokość
drlu
drul
ludr
lurd
rdlu
rdul
uldr
ulrd

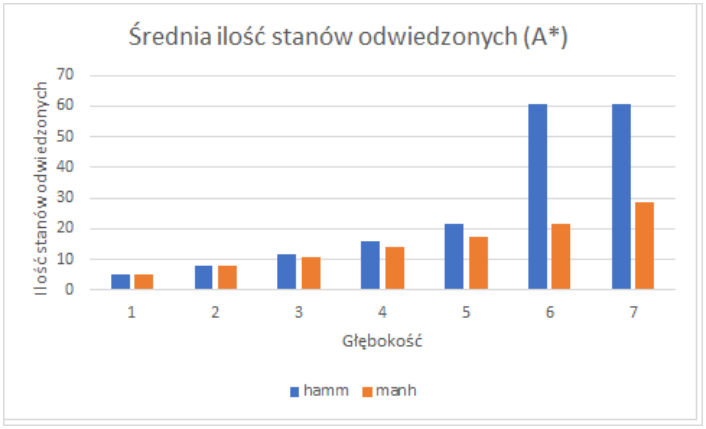

Średnia ilość stanów odwiedzonych (DFS)
Ilość stanów odwiedzonych
900000
800000
700000
600000
500000
400000
300000
200000
100000
0
1
2
3
4
5
6
7
Głębokość
drlu
drul
ludr
lurd
rdlu
rdul
uldr
ulrd

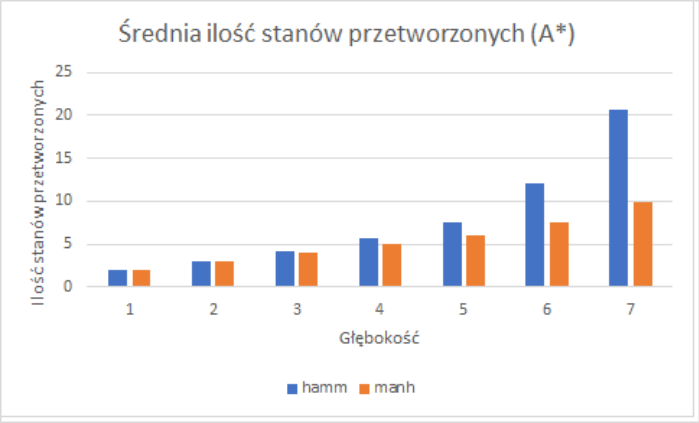

Średnia ilość stanów przetworzonych (DFS)
Ilość stanów przetworzonych
900000
800000
700000
600000
500000
400000
300000
200000
100000
0
1
2
3
4
5
6
7
Głębokość
drlu
drul
ludr
lurd
rdlu
rdul
uldr
ulrd

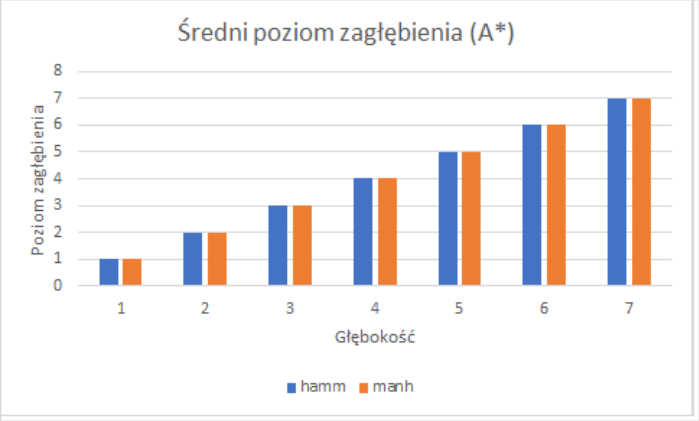

Średni poziom zagłębienia (DFS)
Poziom zagłębienia
25
20
15
10
5
0
1
2
3
4
5
6
7
Głębokość
drlu
drul
ludr
lurd
rdlu
rdul
uldr
ulrd

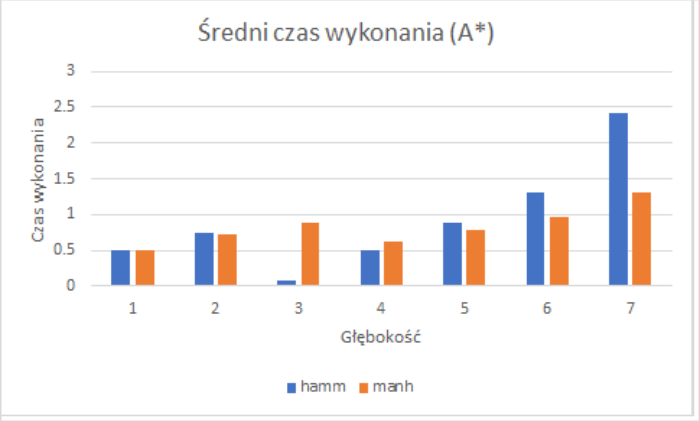

Średni czas wykonania (DFS)
Czas wykonania
7000
6000
5000
4000
3000
2000
1000
0
1
2
3
4
5
6
7
Głębokość
drlu
drul
ludr
lurd
rdlu
rdul
uldr
ulrd


Średnia długość rozwiązania (A*)
Długość rozwiązania
8
7
6
5
4
3
2
1
0
1
2
3
4
5
6
7
Głębokość
hamm
manh


Średnia ilość stanów odwiedzonych (A*)
Ilość stanów odwiedzonych
70
60
50
40
30
20
10
0
1
2
3
4
5
6
7
Głębokość
hamm
manh







6. Dyskusja

Strategia A* okazała się najefektywniejszą ze względu na szybkość odnajdywania rozwiązań. Średni czas wykonania dla poszczególnych głębokości nie przekroczył 2.5 ms. Długości rozwiązań równe były danym ilościom przesunięć.
Średni czas wykonywania jest dużo większy dla DFS niż BFS. Może mieć na to wpływ duża maksymalna głębokość rekursji i sposób napisania algorytmu, w którym zamiast zapisywania układanki dla każdego stanu, zapisujemy jedynie kierunek przesunięcia. Zaoszczędza to pamięć natomiast możliwe, że wpływa niekorzystnie na czas wykonania.
Długość rozwiązania w strategii BFS zależała od głębokości, natomiast w

DFS w dużej mierze od kolejności wyboru stanów. Np. dla układanki o jednym przesunięciu, rozwiązanie mogło być uzyskane już po odwiedzeniu pierwszego z węzłów. Jednak przy innej kolejności wyboru stanów osiągnięto prawie maksymalną głębokość rekursji.
Średni poziom zagłębienia w strategiach BFS i A* równy był ilościom przesunięć w początkowej układance. Natomiast dla DFS w wielu przypadkach dużo większy. Zależało to w od ustawionej maksymalnej głębokości rekursji. W naszym przypadku była to wartość 20.
W strategii A* średnia liczba stanów odwiedzonych i przetworzonych jest nieco niższa dla heurystyki Manhattan.
Średnia ilość stanów odwiedzonych dla BFS sięgała 12000, natomiast dla DFS 800000. BFS odnajdywał rozwiązania po dotarciu do odpowiedniej warstwy, przy czym DFS przeszukiwał rozwiązania na większej głębokości.
BFS okazał się skuteczniejszym algorytmem dla omawianego problemu niż DFS. Nie może się on jednak równać ze strategią A*, która odnalazła rozwiązania w dużo krótszym czasie.

7. Wnioski

1. Strategia A* okazała się najefektywniejsza
2. BFS jest mniej wydajny niż A*. Kolejność wyboru węzłów nie wpływa istotnie na działanie algorytmu.
3. Efektywność DFS w dużej mierze zależy od wyboru maksymalnej głębokości rekursji i kolejności wyboru stanów. Wypadł on jednak najgorzej w tym porównaniu.

Literatura

[1] Mieczysław Muraszkiewicz, Robert Nowak *Sztuczna inteligencja dla inżynierów. Metody ogólne* , Oficyna Wydawnicza Politechniki Warszawskiej, 2022