Data: 13.06.2020
Imię i nazwisko: Karolina Głuszek
Nr albumu: 249034
Nazwa kursu: Projektowanie algorytmów i metody sztucznej inteligencji
Dane prowadzącego: mgr Marta Emirsajłow
Termin zajęć: piątek, 9.15

1. Wprowadzenie

Celem projektu jest analityczne rozwiązanie problemu zbicia czarnego króla białym skoczkiem na zmodyfikowanej planszy do gdy w szachy o rozmiarze 5x5. W tym celu stworzono graf możliwych ruchów skoczka w postaci listy sąsiedztwa, na którym wywołano algorytm przeszukiwania w głąb DFS oraz algorytm znajdowania najkrótszej ścieżki w grafie A*.

2. Opis algorytmu przeszukiwania w głąb

Algorytm przeszukiwania w głąb jest algorytmem stosowanym m.in. do badania spójności grafu, czyli sprawdzenia, czy istnieje ścieżka pomiędzy dwoma wierzchołkami grafu. Działanie algorytmu sprowadza się do przechodzenia zawsze do kolejnego nieodwiedzonego wierzchołka. Jeśli dany wierzchołek nie ma nieodwiedzonych sąsiadów, wracamy do poprzedniego wierzchołka i sprawdzamy jego sąsiadów. Algorytm może zostać zaimplementowany w programie rekurencyjnie lub za pomocą stosu. W przypadku rekurencji wierzchołek oznaczany jest jako odwiedzony, a następnie procedura ta jest rekurencyjnie wywoływana dla każdego nieodwiedzonego sąsiada tego wierzchołka. Dla implementacji ze stosem na początku pierwszy wierzchołek dodawany jest na stos. Następnie dopóki stos nie jest pusty zdejmujemy wierzchołek ze stosu i dodajemy na stos wszystkich jego nieodwiedzonych sąsiadów. Dzięki temu zawsze największy priorytet podczas przechodzenia przez graf mają wierzchołki znajdujące się coraz głębiej. Złożoność czasowa algorytmu wynosi $O(|V| + |E|)$, gdzie $|V|$ to ilość wierzchołków, a $|E|$ ilość krawędzi grafu, ponieważ algorytm przechodzi przez wszystkie wierzchołki i krawędzie. Złożoność pamięciowa wynosi $O(h)$, gdzie h to długość najdłuższej prostej ścieżki w grafie, ponieważ algorytm w każdym momencie działania wymaga zapamiętania jedynie ścieżki od korzenia do bieżącego węzła grafu.

3. Opis algorytmu A*

Algorytm A* to algorytm heurystyczny znajdowania najkrótszej ścieżki w grafie z dowolnego wierzchołka do wierzchołka spełniającego określony warunek. Algorytm wykorzystuje heurystykę do oceny jakości węzłów i wyboru węzła, który w danej chwili należy rozwinąć. Algorytm sprowadza się do minimalizacji funkcji kosztu $f(x) = g(x) + h(x)$, gdzie $g(x)$ to przebyta odległość od wierzchołka początkowego do aktualnie rozważanego w grafie, a $h(x)$ to szacowana odległość, nie większa niż rzeczywista, od aktualnego do wierzchołka docelowego. Algorytm A* jest zawsze optymalny dla danej heurystyki, co oznacza, że nie istnieje inny algorytm, który za pomocą tej samej heurystyki znalazłby krótszą ścieżkę niż A*. Złożoność czasowa algorytmu A* zależy od zastosowanej heurystyki, ale można założyć, że w najgorszym przypadku liczba przeszukanych węzłów rośnie wykładniczo w stosunku do długości najkrótszej ścieżki długości d i wynosi $O(b^d)$, gdzie b to współczynnik rozgałęzienia grafu. Złożoność pamięciowa w najgorszym przypadku również rośnie wykładniczo w stosunku do długości rozwiązania i wynosi $O(b^d)$.

4. Wnioski

W napisanym programie długość ścieżki wyznaczonej przez algorytm przeszukiwania w głąb zależy od kolejności możliwych ruchów skoczka znajdujących się w grafie. Jest to zgodne z założeniami algorytmu, ponieważ stara się on znaleźć jakąkolwiek ścieżkę prowadzącą do zbicia króla, nie zważając na jej długość.

Zaimplementowany algorytm A* zawsze znajduje tę samą optymalną ścieżkę dla swojej heurystyki bez względu na kolejność rozmieszczenia wierzchołków grafu. Mimo to długość znalezionej optymalnej ścieżki wynosi 5 i prowadzi przez pola $W \rightarrow L \rightarrow I \rightarrow R \rightarrow O \rightarrow H$, co nadal nie jest najkrótszą możliwą ścieżką o długości 3 $W \rightarrow L \rightarrow S \rightarrow H$. Można z tego wysnuć wniosek, że wykorzystana funkcja heurystyczna $h(n)$ ustalająca odległość między badanym polem a polem celu jako ich odległość w metryce Manhattan nie jest optymalną funkcją dla opisywanego problemu. Wynika to z faktu, że algorytm zawsze stara się postawić figurę jak najbliżej pola króla, co byłoby optymalne w przypadku bierek poruszających się w linii prostej lub po skosie, ale nie dla skoczka, którego ruchy są bardziej skomplikowane.

5. Źródła

http://algorytmy.ency.pl/artykul/przeszukiwanie_w_glab

http://home.agh.edu.pl/~zobmat/2017/2_tarkowskijakub/teoria/algorytmy.php

https://elektron.elka.pw.edu.pl/~jarabas/ALHE/notatki3.pdf

http://home.agh.edu.pl/~horzyk/lectures/wdi/WDI-Grafy.pdf

https://pl.wikipedia.org/wiki/Algorytm_A*

https://pl.wikipedia.org/wiki/Przeszukiwanie_w_g%C5%82%C4%85b