

05.06.2019r.

Projektowanie Algorytmów i Metody Sztucznej Inteligencji

**Projekt 3 – Implementacja gry „Kółko i Krzyżyk” w
oparciu o algorytm min-max**

Jacek Kopala

1. Wstęp

Celem projektu było zaprojektowanie oraz zaimplementowanie prostej gry wykorzystującej algorytm sztucznej inteligencji. Niniejszy projekt dotyczy gry „Kółko i Krzyżyk” z zastosowaniem w niej algorytmu MinMax, który determinuje zachowanie sztucznej inteligencji jako gracza. Wymaganiami stawianymi wobec projektu była możliwość wyboru wielkości planszy oraz wyboru wymaganej ilości symboli poustawianych w ciągu, które określają wygraną.

2. Algorytm min-max

Algorytm min-max jest jedną z metod stosowaną w celu minimalizacji maksymalnych możliwych strat bądź maksymalizowaniu zysku. Jest to stosunkowo prosta w implementacji technika, która może być zastosowana w grach dwuosobowych gdzie możliwa jest ocena stanu gry w dowolnym momencie. Algorytm wykorzystuje tzw. drzewo gry, które stanowi reprezentację potencjalnych stanów gry po kolejnych ruchach graczy. Bardzo istotnym elementem algorytmu, który w znacznym stopniu definiuje jego skuteczność jest funkcja ewaluacji gry. W rozpatrywanym przypadku, funkcja ewaluacji sprowadza się do określenia wartościowości planszy na której odbywa się rozgrywka. Funkcja ewaluacji określa strategię jaka będzie wykorzystywana do podejmowania kolejnych decyzji w postaci posunięć sztucznej inteligencji. W tym przypadku zostało to zrealizowane poprzez przeszukiwanie całej planszy w kierunku horyzontalnym, wertykalnym, diagonalnym oraz anty-diagonalnym. Program tworzy swoje okno o długości równej ilości znaków wymaganych do wygrania a następnie przesuwane jest po kolejnych komórkach pamięci w sposób wcześniej opisany. Takie podejście umożliwia wykrywanie tzw. open endów oraz close endów – czyli sekwencji które potencjalnie mogą być lub na pewno nie są wzorcem potrzebnym do wygranej. Podejście takie jest zbliżone do algorytmów wyszukiwania wzorca w tekście. Algorytm min-max intensywnie korzysta z funkcji ewaluacji dlatego też to głównie od niej zależy czas wykonywanych obliczeń. Mając już funkcję ewaluacji można przejść do rozpatrywania samego algorytmu min-max. Istotnym aspektem o którym należy wspomnieć, jest fakt, iż jest to algorytm rekurencyjny. Ilość symulowanych (i jednocześnie ewaluowanych) stanów gry także istotnie wpływa na skuteczność działania całości. Z racji, iż zasoby komputera dostępne dla programu są ograniczone, należy określić do jakiej głębokości algorytm będzie symulował kolejne ruchy. Z jednej strony, większa głębokość pozwala na zwiększenie skuteczności algorytmu, z drugiej jednak – większa ilość wymaganych obliczeń obciążona jest większym czasem przetwarzania, co jest istotne w przypadku gier, które powinny działać w czasie niemalże rzeczywistym.

Podsumowując, algorytm wykonywany jest rekurencyjnie tak długo aż gra nie doszła do stanu ustalonego (remis/porażka/wygrana) lub tak długo aż nie został osiągnięty

maksymalny poziom głębokości. Rekurencyjne, naprzemienne symulowanie ruchów graczy pozwala określić ruchy obciążone największym kosztem oraz ruchy zapewniające największy koszt. Na tej podstawie wybierany jest najoptymalniejszy ruch (czyli taki, który albo maksymalizuje zysk, albo minimalizuje koszt).

3. Algorytm min-max

Celem zmniejszenia ilości wykonywanych obliczeń zastosowany został dodatkowo algorytm alfa-beta. Odcięcia alfa-beta pozwalają na pominięcie obliczeń w wypadku gdy obecnie badany stan gry jest przypadkiem gorszym od poprzednio zbadanych. Jako iż program ma na celu minimalizację kosztów/maksymalizację zysków, wybranie opcji jakkolwiek gorszej od wcześniej zbadanych nie jest w żaden sposób uzasadnione. Dlatego też w momencie gdy któraś z gałęzi drzewa gry przyjmuje taką niekorzystną wartość – jest ona pomijana. Odcięcia alfa-beta mogą znacząco wpłynąć na skuteczność całego programu ponieważ czas zaoszczędzony poprzez pomijanie obliczeń gorszych stanów gry może zostać wykorzystany na obliczenia wynikające ze zwiększenia głębokości przeszukiwania drzewa gry.

4. Funkcja wyznaczająca najlepszy ruch

Opisane powyżej algorytmy wywoływane są za pośrednictwem funkcji nadrzędnej, która ma na celu określić najbardziej optymalny ruch sztucznej inteligencji. Funkcja ta przechodzi po całej planszy celem znalezienia pustego pola do którego można wstawić symbol. Po znalezieniu takiej pozycji, ustawiany jest tam symbol, którym gra sztuczna inteligencja oraz wywoływany jest algorytm min-max z odcieczami alfa-beta, po zakończeniu obliczeń proces jest powtarzany aż do momentu przejścia przez wszystkie wolne pola. Największa wartość zwrócona w danym momencie przez algorytm min-max określa najkorzystniejszy ruch i taki też ruch powinien zostać zwrócony jako rezultat działania funkcji.

5. Wnioski

- 5.1. Funkcja ewaluacji definiuje strategię jaką będzie się kierować sztuczna inteligencja. Jej odpowiednie zaprojektowanie pozwala zwiększyć skuteczność wyboru ruchów. Dobrze zaprojektowana funkcja ewaluacji powinna zwiększać określony stan gry tym mocniej im bardziej dany ruch sprzyja wygranej i jednocześnie znacząco go zmniejszać w przypadku zagrożenia przegraną. Początkowo planowane było zwracanie dwóch bardzo dużych pozytywnych wartości stanu gry dla przypadku znajdującego wygraną oraz dla przypadku gdy do wygranej byłby wymagany jeden ruch, analogicznie – dwóch bardzo

dużych, ujemnych wartości dla przegranej i stanu bliskiej przegranej, a także określić wzoru definiującego wartościowanie przypadków open endów. Jednak ze względu na trudność przy doborze wag/wzoru dla open endów wybrana została wersja realizująca tylko cztery w/w skrajne przypadki.

- 5.2.** Algorytm alfa-beta pozwolił na znaczące skrócenie czasu obliczeń co umożliwia rozgrywkę na większych planszach.
- 5.3.** Wraz z ilością wypełnionych miejsc planszy, czas obliczeń maleje – wynika to z faktu iż za każdym ruchem ilość pozostałych kombinacji ruchów spada.
- 5.4.** Zwiększenie głębokości przeszukiwania istotnie wpływa na skuteczność ruchów sztucznej inteligencji, niestety przy większych jej wartościach komputer na którym testowane były algorytmy potrzebował nawet kilku minut na wykonanie obliczeń.

6. Literatura

<https://www.geeksforgeeks.org/minimax-algorithm-in-game-theory-set-3-tic-tac-toe-ai-finding-optimal-move/> [dostęp 01.06.2019]

https://pl.wikipedia.org/wiki/Algorytm_alfa-beta [dostęp 02.06.2019]

<https://zliang.com/pdfs/gomoku.pdf> [dostęp 02.06.2019]

<https://www.youtube.com/watch?v=l-hh51ncgDI> [dostęp 03.06.2019]