

Projektowanie algorytmów i metody sztucznej inteligencji

Projekt 3.

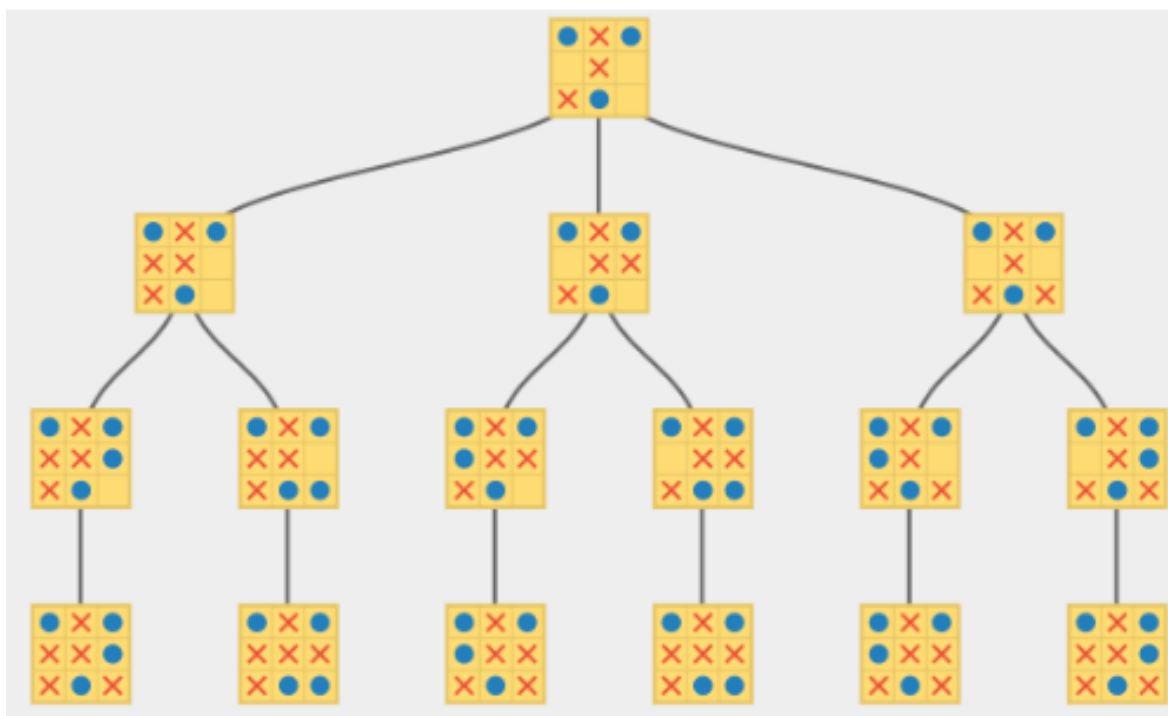
1. Wprowadzenie.

Zadaniem projektowym było wykonanie gry, która wykorzystuje algorytm minimax.

2. Opis algorytmu MINMAX.

Posiadając funkcję S oceniającą wartość stanu gry w dowolnym momencie (gracz min chce ten stan zminimalizować, a gracz max zmaksymalizować), obliczamy drzewo wszystkich możliwych stanów w grze do pewnej głębokości (ograniczonej zazwyczaj przez naszą moc obliczeniową). Zakładając, że rozgałęzienie drzewa stanów jest stałe i wynosi b (czyli na każdy ruch można odpowiedzieć b innymi), a głębokość d (tyle ruchów do przodu symulujemy algorytmem minimax), to mamy b^d stanów końcowych, dla których obliczamy wartość stanu gry funkcją S . Zaczynamy przeglądanie od stanów końcowych, symulując optymalne wybory dla obu graczy, tak aby na głębokości d (w liściach drzewa) była dla nich optymalna liczba S (stan gry po wykonaniu d ruchów). Tak więc gracz min zawsze wybiera ruch, który prowadzi do mniejszej wartości końcowej, a gracz max – przeciwnie. Po przeprowadzeniu tej symulacji gracz, który znajduje się w korzeniu drzewa (aktualnie wykonujący ruch), ma pewność, że jego ruch jest optymalny w kontekście informacji o stanie gry z przeprowadzonej symulacji algorytmem minimax na głębokość d (tzn. maksymalizuje minimalny zysk).

Algorytm służy do wybrania optymalnego ruchu w danym momencie, dlatego po ruchu przeciwnika musimy przeprowadzić symulację ponownie. Większa głębokość d symulacji prowadzi do lepszych ruchów. Optymalizacja algorytmu z odcięciami alfa-beta pozwala, w optymalnym przypadku, zmniejszyć ilość rozpatrywanych stanów, co w efekcie pozwala nam symulować ruchy prawie dwa razy głębiej.



Rysunek 1. Przykład drzewa gry dla planszy 3x3

3. Opis gry kółko i krzyżyk

W pierwszej kolejności gracz wybiera, czy zaczyna on, czy komputer. Następnie poprzez wybór numeru ID pola stawia we wskazanym miejscu krzyżyk.

```

-----
                                Kolko krzyzyk
-----

Czy chcesz zaczac? (t/n) : t

Wybierz numer pola

      1 | 2 | 3
      ---
      4 | 5 | 6
      ---
      7 | 8 | 9

Wolne pola : 1 2 3 4 5 6 7 8 9

Wybierz pole = 

```

Następnie komputer dzięki algorytmowi MINMAX wybiera najbardziej korzystną dla niego i jednocześnie najgorszą możliwość dla nas. W momencie kiedy w linii są tylko kółka wygrywa komputer, w momencie kiedy krzyżyki są w linii wygrywa gracz. W sytuacji gdy nikomu nie udało się ustawić w linii, Gra kończy się bez zwycięzcy.

```
postawiles X na polu 4

  O | O |
  ---
  X | X |
  ---
  | X |

KOMPUTER postawil 0 na polu 3

  O | O | O
  ---
  X | X |
  ---
  | X |

KOMPUTER wygral

Czy chcesz wyjsc?(t/n) : 
```

4. Wnioski

Algorytm radzi sobie bardzo dobrze w rozgrywce. Niekiedy można odczuć powtarzalność wyko-nywanych ruchów, szczególnie w początkowych etapach rozgrywek. Rozgrzywka dwóch algorytmówgrających przeciwko sobie za każdym razem wygląda jednakowo. Złożoność algorytmu jest dosyćduża, ze względu na ilość ruchów, które algorytm ma do rozpatrzenia. Na moim laptopie gra prze-staje być grywalna przy przewidywaniu 8 następnych ruchów, a najprzyjemniej gra się przy około 6. Algorytm nie jest niepokonany, skuteczność jego działania jest uzależniona od tego, jak dokładniefunkcja heurystyczna przybliża dany system.