

# Gry dwuosobowe algorytm minimaxowy i przycinanie alfa-beta

Joanna Kołodziejczyk

## 1 Problem do rozwiązania

Celem laboratorium jest zapoznanie się z algorytmami strategii dla gier dwuosobowych na przykładzie kółka-krzyżyk.

## 2 Algorytm minimaxowy

Algorytm minimaxowy może być użyty do zaimplementowania optymalnej gry w każdej deterministycznej, dwuosobowej grze pełnej informacji. W praktyce jest zachłanny i nie nadaje się dla gier z dużą liczbą stanów.

Działanie algorytmu można przedstawić: Biorąc pod uwagę stan gry, algorytm oblicza wartości (wartość wygranej, przegranej, remisu) dzieci z danego stanu i wybiera ten, który ma wartość maksymalną, jeśli jest to kolej Maxa, i ten, który ma wartość minimalną, jeśli jest to kolej Mina.

Algorytm może być zaimplementowany przy użyciu poniższych funkcji rekurencyjnych odpowiednio dla węzłów Max i Min.

```
1: max_value(node):
2:     if end_state(node): return value(node)
3:     v = -Inf
4:     for each child in node.children():
5:         v = max(v, min_value(child))
6:     return v
```

```
1: min_value(node):
2:     if end_state(node): return value(node)
3:     v = +Inf
4:     for each child in node.children():
5:         v = min(v, max_value(child))
6:     return v
```

## 2.1 Zadanie 1 (pkt 2)

Założmy, że na planszy kółka krzyżyk pojawia się następujący układ, po tym jak gracz  $X$  wstawił swój krzyżyk:

```
o | x | o
x |   |
x | o |
```

Teraz kolej, aby zagrał gracz  $O$ . Oceń wartość tego stanu gry, jak również innych stanów w drzewie gry, gdzie powyższa pozycja jest korzeniem, używając algorytmu minimaxowego. Rozpocznij rekurencję, wywołując  $\text{max\_value}(node)$  jeżeli grasz jako gracz  $O$  lub  $\text{min\_value}(node)$ , jeżeli grasz jako gracz  $X$ , gdzie  $node$  jest powyższą pozycją planszy.

Co powinno zawierać rozwiązanie?

1. Jaki pierwszy ruch powinien wykonać wybrany gracz.
2. Prezentację zmiany wartości  $v$

## 3 Algorytm przycinanie alfa-beta

W przypadku małych drzew algorytm można stosować bez funkcji heurystycznej, a w przypadku dużych drzew można połączyć z oceną heurystyczną, która pozwala nie rozwijać drzewa do końca.

Aby zaimplementować przycinanie alfa-beta w podobny sposób jak algorytm minimaxowy wprowadza się niewielkich zmiany w funkcjach  $\text{min\_value}$  i  $\text{max\_value}$ .

```
1: max_value(node, alpha, beta):
2:     if end_state(node): return value(node)
3:     v = -Inf
4:     for each child in node.children():
5:         v = max(v, min_value(child, alpha, beta))
6:         alpha = max(alpha, v)
7:         if alpha >= beta: return v
8:     return v
```

```
1: min_value(node, alpha, beta):
2:     if end_state(node): return value(node)
3:     v = +Inf
4:     for each child in node.children():
5:         v = min(v, max_value(child, alpha, beta))
6:         beta = min(beta, v)
7:         if alpha >= beta: return v
8:     return v
```

Ważne jest, aby pamiętać, że wartość alfa jest aktualizowana tylko w węzłach Max, a wartość beta jest aktualizowana tylko w węzłach Min. Zaktualizowane wartości są przekazywane jako argumenty w dół do dzieci, ale nie do wywołującego węzła macierzystego. (To znaczy, że argumenty są przekazywane jako wartości, a nie jako referencje).

Interpretacja alfa i beta: podają one przedział możliwych wartości w węźle, który jest przetwarzany:  $alpha \leq value \leq beta$ . Interwał ten jest aktualizowany w trakcie działania algorytmu, a jeśli w pewnym momencie interwał skurczy się tak, że  $alpha = beta$ , znamy  $value$  i możemy zwrócić ją do węzła macierzystego bez przetwarzania kolejnych węzłów. Może się również zdarzyć, że  $alpha > beta$ , co oznacza, że obecny węzeł nigdy nie zostanie odwiedzony w optymalnej grze, a jego przetwarzanie może zostać przerwane.

Przy uruchamianiu rekurencji w węźle głównym używamy minimalnej i maksymalnej wartości gry jako wartości odpowiednio  $alpha$  i  $beta$ . Na przykład dla kółka krzyżyk i szachów, gdzie wynikiem jest wygrana (1) /przegrana (-1), jest to  $alpha = -1$  i  $beta = 1$ . Jeżeli zakres możliwych wartości nie jest z góry określony, inicjalizujemy je jako  $alpha = -\infty$ ,  $beta = +\infty$

### 3.1 Zadanie 2 (pkt 3)

Załóżmy, że na planszy kółka krzyżyk pojawia się następujący układ, po tym jak gracz  $X$  wstawił swój krzyżyk:

```

o | x | o
x |   | x
 | o |

```

W grze jest kolej, aby zagrał gracz  $O$ . Oceń wartość tego stanu gry, jak również innych stanów w drzewie gry, gdzie powyższa pozycja jest korzeniem, używając algorytmu przycinania alfa-beta. Rozpocznij rekurencję, wywołując  $max\text{-value}(node, -1, 1)$  jeżeli grasz jako gracz  $O$  lub  $min\text{-value}(node, -1, 1)$ , jeżeli grasz jako gracz  $X$ , gdzie  $node$  jest powyższą pozycją planszy.

Co powinno zawierać rozwiązanie?

1. Jaki pierwszy ruch powinien wykonać wybrany gracz.
2. Prezentację zmiany wartości  $v$
3. Prezentację zmiany wartości  $\alpha$  i  $\beta$

### 3.2 Przekazanie zadań

Rozwiązania proszę podpisać w Teamsach. Plik nazwać Nazwisko.Imię.pdf