

Temat: Kółko i krzyżyk – dwuosobowe gry deterministyczne

Implementacja

W ramach projektu dokonano implementacji algorytmu minima oraz algorytmu minima z przycinaniem alfa-beta. Algorytm minimax rekurencyjnie wyznacza ocenę danego stanu. Jest to algorytm rekurencyjny, który działa do zadanego poziomu w głąb drzewa. Gracz pierwszy jest graczem max, który próbuje maksymalizować tę funkcję oceny i wybrać stan o jej największej sumie, a min odwrotnie minimalizuje funkcję oceny i wybiera stan o jej najmniejszej sumie. W przypadku, gdy głębokość drzewa nie jest maksymalna i w danej gałęzi nie wystąpi stan terminalny, na którego podstawie da się stanowi przypisać funkcję oceny, przypisuje mu się wartość zero.

Algorytm z włączonym przycinaniem alfa-beta charakteryzuje się większą szybkością i mniejszą liczbą przeszukanych stanów. Dla gracza max usuwane są ścieżki gdy funkcja oceny jest gorsza niż obecnie najlepsza dla innej ścieżki na tym samym poziomie. Analogicznie jest dla gracza min.

Wykorzystana heurystyka

3	2	3
2	4	2
3	2	3

Konfiguracje CPU-1 vs CPU-2	CPU-1 wygrał (True, False)	remis	CPU-1 l. przeszukanych stanów	CPU-2 l. przeszukanych stanów
(D=9, ab=False) vs (D=9, ab=False)		x	557487	60688
(D=9, ab=False) vs (D=9, ab=True)		x	557487	6585
(D=9, ab=False) vs (D=3, ab=False)		x	557487	574
(D=3, ab=False) vs (D=9, ab=True)	False		929	9274
(D=3, ab=False) vs (D=3, ab=False)	False		929	570
(D=3, ab=False) vs (D=3, ab=True)	False		929	570
(D=1, ab=False) vs (D=3, ab=False)	False		21	563
(D=2, ab=False) vs (D=3, ab=False)		x	165	574
(D=3, ab=False) vs (D=3, ab=False)	False		929	570
(D=4, ab=False) vs (D=3, ab=False)		x	4785	574
(D=9, ab=False) vs (D=3, ab=False)		x	557487	574
(D=1, ab=False) vs (D=0, ab=False)	D=0 nie jest obsługiwany przez algorytm minimax()			

Z przeprowadzonych eksperymentów wynika, że dwa AI posiadające tę samą głębokość przeszukiwania ze sobą zremisują. Włączenie przycinania alfa-beta spowoduje, że znacząco zmniejszy się czas oczekiwania algorytmu na wykonanie ruchu (mniej stanów zostanie przeszukanych). Dodatkowo zauważono, że dla głębokości równej 3 algorytm działa już całkiem dobrze. Gdy gra z przeciwnikiem, którego głębokość jest równa 9 potrafi zremisować. Umie wygrać będąc graczem drugim tylko z przeciwnikiem, którego D=1 (nie przeszukuje drzewa tylko ruchy opiera na heurystyce). Ciekawe jest to, że gdy gra sam ze sobą to udaje mu się wygrać. Ustawienie, które w tym przypadku jest krytyczne to:

2		
	1	

Ten stan prowadzi do stanu terminalnego dla gracza dwa. Jest to ciekawa rozgrywka, ponieważ okazuje się, że głębokość 3 pozwala na wybranie pozornie

2	1	
	1	
	2	

optymalnych ruchów, które w konsekwencji prowadzą jednego stanu terminalnego, w którym gracz drugi zawsze wygra.

2	1	1
	1	
2	2	

Dla $D=0$ funkcja `minimax()` się nie odpali. Gracza o $D=0$ odpalamy przy użyciu funkcji `tictactoe_rd()`, traktując jako gracza losowego.

Przeprowadzając eksperymenty dla AI z AI losującym okazało się, że nawet AI z $D=1$ bardzo często wygrywa z AI losującym. Poniższa tabela reprezentuje uzyskane wyniki z uśrednionych 100 gier.

Głębokość D	Alfa-beta = False				Alfa-beta = True			
	CPU1- win	CPU1- lost	tie	Śr. przeszukane stanów	CPU1- win	CPU1- lost	tie	Śr. przeszukane stanów
1	100	0	0	19	97	3	0	18
2	95	3	2	154	96	0	4	154
3	98	0	2	884	97	0	3	875
4	99	0	1	4708	100	0	0	2791
8	99	0	1	429098	100	0	0	47223

Jest istotna różnica w ilości przeszukiwanych stanów wraz ze wzrostem D . Na podstawie powyższych eksperymentów można zauważyć, że AI losujący nie jest skuteczny. Wyniki pokazują też, że dla $D \leq 3$ przycinanie alfa-beta nie ma większego znaczenia w przeciwieństwie do większych głębokości. Warto zaznaczyć, że algorytm `minimax` działa najlepiej dla optymalnych ruchów przeciwnika a nie losowych.

Czy zaczynając zawsze tak samo (i z tymi samymi ustawieniami) przebieg rozgrywki jest deterministyczny?

Tak, zaczynając tak samo z tymi samymi ustawieniami przebieg rozgrywki jest deterministyczny.

Czy można wygrać z komputerem? Jeżeli tak to kiedy?

Przy głębokości równej 0 algorytm bazuje na dostarczonej heurystyce dla gry 3x3, czyli znacznie chętniej wybiera środek i mniej chętnie rogi. Przy głębokości wynoszącej 1 algorytm przewiduje tylko swój jeden ruch. Nie zauważa on ruchów, które mogą doprowadzić do wygranej przeciwnika. Nadal jest dosyć łatwo z nim wygrać. Przy głębokości 2 algorytm umie blokować ruchy przeciwnika, które w prosty sposób prowadzą do wygranej. Na tym etapie przy niedogodnym rozpoczęciu algorytm nie umie wygrać z przeciwnikiem (graczem, niezbyt inteligentnym AI). Przy większych głębokościach wykonywanie nieoptymalnych ruchów skutkuje co najmniej remisem, a przy pomyłce gracza przegraną.

Czy jeżeli wiemy, że przegramy/zremisujemy to czy taka gra może sprawiać przyjemność?

Tak, gdy potrafimy docenić i czerpać przyjemność z samego przebiegu gry albo dobre towarzystwo do wspólnego grania.