

WSI22Z

Raport z ćwiczenia laboratoryjnego nr 3

Jan Kuc, 303753

1. Cel eksperymentów

Zadanie polega na implementacji algorytmu minimax z obcinaniem $\alpha - \beta$, a następnie przetestowanie go na grze **Pick**. Dla różnych ruchów o tej samej jakości, algorytm powinien zwracać losowy z nich.

Do realizacji zadania wykorzystano gotową implementację gry **Pick34** z repozytorium dołączonym do polecenia zadania.

2. Opis eksperymentów i decyzje projektowe

Na potrzeby implementacji algorytmu minima z obcinaniem $\alpha - \beta$ oraz metod pomocniczych, stworzono klasę **PlayPick**, która jako argumenty inicjalizujące przyjmuje maksymalne głębokości przeszukiwania dla gracza MIN i gracza MAX.

Klasa symuluje rozgrywkę między dwoma wcześniej wspomnianymi graczami, korzystając z klasy **Pick** implementującej interfejs klasy **Game**. Dzięki temu, możliwa jest symulacja rozgrywki pomiędzy stroną minimalizującą i maksymalizującą z różnymi głębokościami poszukiwania.

Przyjęta heurystyka działa dopiero, kiedy gracz ma już wybrane **n-1** numerów, gdzie **n** oznacza liczbę numerów z których może zostać zsumowana wartość docelowa gry – w badanym przypadku gry **Pick34**, $n=4$, wartość docelowa równa 34.

Przyjęto, że gracz **MAX** zaczyna.

Ponadto, należało zastosować w algorytmie własną implementację funkcji heurystycznej, która jest używana do oceny jakości ruchu, w przypadku kiedy wyczerpano głębokość przeszukiwania i ruch znajduje się poza *depth_max*.

Jako funkcję heurystyczną przyjęto różnicę między liczbą pojedynczych, wolnych numerów, którymi aktualny gracz może wygrać grę w kolejnym ruchu, a odpowiadającą liczbą wolnych numerów dla przeciwnego gracza. Dla gracza **MIN** zastosowano odwrócenie znaku.

W przypadku odnalezienia, podczas przeszukiwania kolejnych możliwości ruchu, stanu terminalnego, do węzła zwracane są wartości:

- 1000 – w przypadku wygranej gracza **MAX**
- 0 – w przypadku remisu
- -1000 – w przypadku wygranej gracza **MIN**

Jeśli jest to ewaluacja końcowa gry, mająca wyłonić ostateczny rezultat, zwracane wartości są dokładnie takie jak je zaprezentowano. Jednakże, jeśli stan terminalny zostanie odnaleziony jeszcze podczas szukania najlepszego ruchu dla danej strony w trakcie gry,

zwracana wartość jest modyfikowana o liczbę ruchów dzielącą aktualny stan od terminalnego. Takie podejście gwarantuje rozróżnienie jakości wygranej w 1 lub więcej ruchach.

W implementacji uwzględniono również możliwość ustawienia danemu graczowi głębokości przeszukiwania mniejszej od 0, co skutkuje w wykonywaniu całkowicie losowych ruchów przez wybraną stronę.

3. Oczekiwane rezultaty

W trakcie eksperymentów, oprócz ostatecznego wyniku, badano również rozmiar przestrzeni poszukiwań dla każdego z graczy, z tego względu, iż przed każdą symulacją rozgrywki, wybierana jest głębokość przeszukiwania każdej ze stron, więc liczba stanów do których dotrze algorytm będzie różna.

Spodziewana jest raczej przewaga zwycięstw strony, która w danym przypadku będzie miała ustawioną większą głębokość przeszukiwań, gdyż algorytm może wtedy dotrzeć do większej ilości stanów terminalnych, a więc na wcześniejszym etapie poznać ostateczne wyniki dla większej ilości ścieżek.

Z uwagi na wymaganie dotyczące losowego wyboru ruchu, w przypadku wystąpienia kilku ruchów o takiej samej ocenie jakości, wyniki symulacji dla takich samych ustawień początkowych algorytmu, będą się różnić.

4. Prezentacja i omówienie wyników

Tabela 1 Prezentacja wyników symulacji gier.

Liczba gier	Depth MAX	Depth MIN	Wygrane MAX	Remisy	Wygrane MIN	Przesz. stany MAX (avg)	Przesz. stany MIN (avg)
1000	4	4	621	0	379	4191.495	4763.995
200	5	2	126	0	74	28687.49	185.77
200	1	4	116	0	84	61.04	5172.65
200	6	6	130	0	70	104395.085	203882.345
200	losowo	4	102	6	92	5.615	4970.51
200	4	losowo	135	5	60	4149.02	4.77
200	5	3	123	0	77	29218.51	1020.29

5. Wnioski i obserwacje

Wyniki zaprezentowane w Tabeli 1 pokazują, przewagę w liczbie wygranych gier po stronie gracza MAX, co najprawdopodobniej związane jest z przyjętym założeniem, że to właśnie ten gracz rozpoczyna gry.

Z drugiej strony, dla głębokości przeszukiwań MAX=1 i MIN=4 nadal wystąpiła przewaga w liczbie wygranych symulacji dla gracza MAX, co nie jest oczekiwanym rezultatem, lecz może wynikać z przyjętej heurystyki.

Jeśli chodzi o liczby przeszukiwanych stanów, otrzymano raczej spodziewane wyniki. Rozmiar przestrzeni przeszukiwań dla obu stron wzrasta wraz ze zwiększaniem głębokości przeszukiwań.

Przycinanie $\alpha - \beta$ nie zmienia decyzji komputera, jeśli chodzi o wykonanie ruchu, a znacznie zmniejsza liczbę przeszukiwanych stanów.

Nawet dla przypadków, gdzie występuje przewaga w głębokości przeszukiwania po stronie MIN, za każdym razem większa liczba gier kończy się wygraną MAX. Wskazywałoby to na startowanie gry z przewagą po stronie rozpoczynającej.

Przeprowadzono test dla takich samych głębokości przeszukiwań graczy MAX i MIN jak w ostatnim wierszu Tabela 1, lecz zmieniono stronę rozpoczynającą. Oto jak prezentują się wyniki eksperymentu:

Tabela 2 Test przeprowadzony po zamianie strony zaczynającej grę.

Liczba gier	Depth MAX	Depth MIN	Wygrane MAX	Remisy	Wygrane MIN	Przesz. stany MAX (avg)	Przesz. stany MIN (avg)
200	5	3	73	9	118	15492.01	896.945

Wyniki umieszczone w Tabela 2 potwierdzają hipotezę, że w badanej grze, wiele zależy od tego która strona rozpoczyna rozgrywkę – gracz rozpoczynający ma przewagę.

Rezultaty testów, szczególnie dla przypadków z losowym wyborem ruchów dla gracza, wskazują, że wybrana heurystyka może nie dawać wystarczająco dobrych efektów. Stąd występowanie zwycięstw strony losującej kolejne ruchy.