

# Inteligencja Obliczeniowa w Analizie Danych Cyfrowych (IOwADC) - Projekt I

Robert Barcik, Konrad Bodzioch, Dominik Breksa

10/03/2024

## 1 Cel Ćwiczenia

Analiza oraz pełna implementacja probabilistycznego wariantu gry *Nim* (nazywanego dalej *Nimby*). Porównanie złożoności czasowej, czasu podejmowania decyzji przez różne algorytmy AI. Testowane algorytmy to: *Negamax*, *Negamax bez predykcji Alpha*, *Beta*, *Negamax dostosowanego do Nimby*. Modyfikacja oryginalnego algorytmu z biblioteki *EasyAI - Negamax* na wariant poprawnie przewidujący ruchy dla wariantu gry *Nimby*.

## 2 Opis Problemu

Wariant gry *Nimby* zawiera małą szansę 10%, że gracz nie wykona ruchu w sposób zamierzony (weźmie o 1 element więcej ze stosu niż planował). Skutkuje to niedostosowaniem algorytmu AI - nie jest on w stanie przewidzieć w najlepszym ruchu, i podejmie złą decyzję.

Wariant gry *Nimby*, nie może być użyty do samej predykcji ruchu przez AI. Ponieważ algorytm AI mógłby budować drzewo stanów gry błędnie z uwagi na to, że wykonanie ruchu nie jest deterministyczne. A więc bardziej ilustratywnie jak dany Algorytm AI przechodzi przez drzewo stanów, to przy jednym przejściu mógłby wyliczyć dany ruch na np. 100, a przy ponownym przejściu na 200. Problem staje się jeszcze większy przy zastosowaniu zapisów stanów gry dla zmniejszenia obliczeń, ponieważ błąd mógłby się powzielić wielokrotnie, z uwagi na zapisanie błędnej wartości oszacowania ruchu.

Problematyczna jest też ilość obliczeń (złożoność obliczeniowa), ponieważ charakterystyka konkretnej rozważanej gry (losowej lub nie), informuje nas o nie wielomianowości złożoności obliczeniowej. Problem się dodatkowo nasila, ponieważ mamy do dyspozycji język Python (znany z powolności).

## 3 Realizacja Rozwiązania

Poniżej przedstawimy sposoby podjęte, do rozwiązania danych problemów. Należy zauważyć, że dla każdego z nich wykorzystano tak zwane tabele tranzycji, które zapisywały należny stan gry, tak aby przyspieszyć znacząco obliczenia.

### 3.1 Nim

Gra w której dzieli się elementy na jakieś ich stosy. Gracze na zmianę biorą niezerową liczbę elementów z jednego ze stosów. Przegrywa gracz, który musi wziąć ostatni element. W naszym modelu gry przyjęliśmy, 4 stosy, każdy po 5 elementów. Domyślnie zaczyna gracz o numerze 1, następnie kolejkę otrzymuje gracz o id 2.

### 3.2 Nimby

Stworzenie probabilistycznego modelu gry Nimby. Uzyskano satysfakcjonujący efekt poprzez użycie obiektowego polimorfizmu. W języku Python taka metodologia pozwoliła stworzyć prawie identyczną wersję gry Nimby względem oryginalnej gry Nim (zmniejszając błąd powstały z uwagi na dużą rozbieżność klas).

Sam efekt losowego wpływu na przebieg rozgrywki uzyskano poprzez nadpisanie metody "make\_move", w której dodano losowość, z wykorzystaniem trywialnie biblioteki random z języka Python. Dodano również wskazówki do typów dla czytelności.

Aby wyeliminować błędy w czasie pomiaru wpływu nie determinizmu na stan wygranych, powtórzono pomiary 8 razy. Powtórki zastosowano, też dla zmiany gracza rozpoczynającego.

### 3.3 Adresacja problemu błędnego budowania drzewa stanów gry przez algorytm AI

Specyfika budowy biblioteki EasyAI, powoduje, że AI, podejmuje decyzję o ruchu bazując na tym samym obiekcie, w którym dzieje się sama gra. Jest to problematyczne, ponieważ tak jak wymieniono w opisie problemu mogą powstać błędy w jego działaniu. zaadresowaliśmy ten problem za pomocą odpowiedniej izolacji stanu gry. W skrócie z uwagi na to, że Python jest bardzo dynamicznym językiem to podczas interpretacji programu dynamicznie podmieniliśmy skład metod w klasie AI.Player (nie musząc tym samym tworzyć polimorfizmu). Podmianę zrobiliśmy na metodę w zasadzie identyczną, ale sam obiekt gry jest modyfikowany w locie i odpowiednio bindowany do oryginalnej metody "make\_move" z gry Nim. Oszczędza to niepotrzebne alokacje, które mogły by być niepożądane.

### 3.4 Negamax

Za pomocą algorytmu Negamax zostają rozegrane różne gry zarówno Nim (wariant deterministyczny) jak i Nimby (wariant probabilistyczny). Próby zostały również przeprowadzone dla algorytmów Negamax bez alpha / beta pruning - u oraz algorytmu Negamax dla wersji probabilistycznej.

### 3.5 Negamax bez alpha / beta pruning - u

Jest to bardzo prymitywna wersja algorytmu Negamax, nie zawiera szacowań alpha / beta. Algorytm zawsze przechodzi przez wszystkie dostępne ruchy, dla danego stanu gry, po drodze zapisuje je, aby uniknąć powtórnych obliczeń. Informacja alpha / beta nie jest przetwarzana, lecz porównywany jest tylko z najlepszym dotychczas znalezionym ruchem. Algorytm jest bardzo wyszczuplony względem oryginału, nie mniej jednak sporo wolniejszy.

W kodzie Klasa implementująca ten algorytm to: "NoAlphaBetaNegamax".

### 3.6 Negamax dla probabilistycznego wariantu gry

Algorytm dostosowany do gry nimby jest również bardzo podobny do wariantu klasycznego, nie mniej jednak z małą modyfikacją. Problem lepszego szacowania wartości danego stanu gry podjęto poprzez rozdzielenie każdego ruchu na 2 pod-ruchy, pierwszy z wagą 0.9, tak jakby ruch miał postąpić zgodnie z klasyczną intencją algorytmu oraz z wagą 0.1, tak jakby miał nastąpić przypadek losowy. Następnie dodano obie wagi i aby zmniejszyć błędy numeryczne nie dzielono sumy przez 2 (aby np. wyciągnąć średnią).

W kodzie Klasa implementująca ten algorytm to: "ExpectedNegamax".

## 4 Wyniki obliczeń

Poniżej przedstawiono wyniki zebranych danych dla wszystkich kombinacji w wersji tabelarycznej. Do celu agregacji danych wykorzystano bibliotekę pandas.

	Typ gry	Algorytm	Głębokości algorytmu graczy	Rozpoczynający	Wygrany	Czas	Rundy	Średni czas rundy
0	Nim	Negamax	[14 7]	1	1	0.318325	14	0.022738
1	Nim	Negamax	[4 3]	1	2	0.114342	17	0.006726
2	Nim	Negamax	[14 7]	2	1	0.456387	17	0.026846
3	Nim	Negamax	[4 3]	2	1	0.065902	17	0.003877
4	Nimby	Negamax	[14 7]	1	1	0.313044	18	0.017391
5	Nimby	Negamax	[14 7]	1	1	0.517039	16	0.032315
6	Nimby	Negamax	[14 7]	1	1	0.236838	14	0.016917
7	Nimby	Negamax	[14 7]	1	1	0.253878	16	0.015867
8	Nimby	Negamax	[14 7]	1	2	0.251546	17	0.014797
9	Nimby	Negamax	[14 7]	1	1	0.234277	16	0.014642
10	Nimby	Negamax	[14 7]	1	1	0.236962	18	0.013165
11	Nimby	Negamax	[14 7]	1	1	0.267444	16	0.016715
12	Nimby	Negamax	[4 3]	1	2	0.049637	19	0.002612
13	Nimby	Negamax	[4 3]	1	2	0.050136	19	0.002639
14	Nimby	Negamax	[4 3]	1	1	0.046950	18	0.002608
15	Nimby	Negamax	[4 3]	1	1	0.049760	18	0.002764
16	Nimby	Negamax	[4 3]	1	2	0.056290	19	0.002963
17	Nimby	Negamax	[4 3]	1	2	0.050018	19	0.002633
18	Nimby	Negamax	[4 3]	1	1	0.046325	18	0.002574
19	Nimby	Negamax	[4 3]	1	2	0.053836	21	0.002564
20	Nimby	Negamax	[14 7]	2	1	0.266943	17	0.015703
21	Nimby	Negamax	[14 7]	2	1	0.255594	17	0.015035
22	Nimby	Negamax	[14 7]	2	2	0.271558	18	0.015087
23	Nimby	Negamax	[14 7]	2	2	0.267713	18	0.014873
24	Nimby	Negamax	[14 7]	2	1	0.262679	17	0.015452
25	Nimby	Negamax	[14 7]	2	1	0.292812	17	0.017224
26	Nimby	Negamax	[14 7]	2	1	0.284414	19	0.014969
27	Nimby	Negamax	[14 7]	2	1	0.255004	17	0.015000
28	Nimby	Negamax	[4 3]	2	1	0.043430	17	0.002555
29	Nimby	Negamax	[4 3]	2	1	0.047759	19	0.002514
30	Nimby	Negamax	[4 3]	2	1	0.047548	19	0.002503
31	Nimby	Negamax	[4 3]	2	2	0.049752	20	0.002488
32	Nimby	Negamax	[4 3]	2	2	0.056137	20	0.002807
33	Nimby	Negamax	[4 3]	2	1	0.047712	19	0.002511
34	Nimby	Negamax	[4 3]	2	1	0.041717	17	0.002454
35	Nimby	Negamax	[4 3]	2	2	0.044023	18	0.002446
36	Nim	NoAlphaBetaNegamax	[14 7]	1	1	0.619814	18	0.034434
37	Nim	NoAlphaBetaNegamax	[4 3]	1	1	0.212269	18	0.011793
38	Nim	NoAlphaBetaNegamax	[14 7]	2	1	0.620284	19	0.032647
39	Nim	NoAlphaBetaNegamax	[4 3]	2	1	0.165392	17	0.009729
40	Nimby	NoAlphaBetaNegamax	[14 7]	1	2	0.608329	23	0.026449
41	Nimby	NoAlphaBetaNegamax	[14 7]	1	1	0.646876	18	0.035938
42	Nimby	NoAlphaBetaNegamax	[14 7]	1	2	0.437430	19	0.023023
43	Nimby	NoAlphaBetaNegamax	[14 7]	1	1	0.455317	20	0.022766
44	Nimby	NoAlphaBetaNegamax	[14 7]	1	1	0.453267	20	0.022663
45	Nimby	NoAlphaBetaNegamax	[14 7]	1	2	0.507788	19	0.026726
46	Nimby	NoAlphaBetaNegamax	[14 7]	1	1	0.466160	22	0.021189
47	Nimby	NoAlphaBetaNegamax	[14 7]	1	2	0.427578	19	0.022504
48	Nimby	NoAlphaBetaNegamax	[4 3]	1	1	0.143071	18	0.007948
49	Nimby	NoAlphaBetaNegamax	[4 3]	1	1	0.142174	18	0.007899
50	Nimby	NoAlphaBetaNegamax	[4 3]	1	1	0.135833	18	0.007546
51	Nimby	NoAlphaBetaNegamax	[4 3]	1	2	0.143845	19	0.007571
52	Nimby	NoAlphaBetaNegamax	[4 3]	1	1	0.146557	18	0.008142
53	Nimby	NoAlphaBetaNegamax	[4 3]	1	2	0.147836	21	0.007040
54	Nimby	NoAlphaBetaNegamax	[4 3]	1	1	0.157358	20	0.007868
55	Nimby	NoAlphaBetaNegamax	[4 3]	1	1	0.140537	18	0.007808
56	Nimby	NoAlphaBetaNegamax	[14 7]	2	2	0.482979	20	0.024149
57	Nimby	NoAlphaBetaNegamax	[14 7]	2	2	0.427284	20	0.021364
58	Nimby	NoAlphaBetaNegamax	[14 7]	2	2	0.423783	20	0.021189
59	Nimby	NoAlphaBetaNegamax	[14 7]	2	2	0.432718	20	0.021636
60	Nimby	NoAlphaBetaNegamax	[14 7]	2	1	0.423272	21	0.020156
61	Nimby	NoAlphaBetaNegamax	[14 7]	2	1	0.483813	21	0.023039
62	Nimby	NoAlphaBetaNegamax	[14 7]	2	2	0.483791	22	0.021991
63	Nimby	NoAlphaBetaNegamax	[14 7]	2	1	0.444033	23	0.019306
64	Nimby	NoAlphaBetaNegamax	[4 3]	2	1	0.125598	19	0.006610
65	Nimby	NoAlphaBetaNegamax	[4 3]	2	1	0.153793	21	0.007323
66	Nimby	NoAlphaBetaNegamax	[4 3]	2	1	0.113963	17	0.006704
67	Nimby	NoAlphaBetaNegamax	[4 3]	2	2	0.139847	20	0.006992
68	Nimby	NoAlphaBetaNegamax	[4 3]	2	1	0.127867	19	0.006730
69	Nimby	NoAlphaBetaNegamax	[4 3]	2	2	0.151864	22	0.006903
70	Nimby	NoAlphaBetaNegamax	[4 3]	2	1	0.156149	21	0.007436
71	Nimby	NoAlphaBetaNegamax	[4 3]	2	1	0.116710	17	0.006865
72	Nim	ExpectedNegamax	[14 7]	1	1	4.401969	18	0.244554
73	Nim	ExpectedNegamax	[4 3]	1	2	0.103094	17	0.006064
74	Nim	ExpectedNegamax	[14 7]	2	1	2.847942	17	0.167526
75	Nim	ExpectedNegamax	[4 3]	2	1	0.096010	17	0.005648
76	Nimby	ExpectedNegamax	[14 7]	1	1	4.254633	20	0.212732
77	Nimby	ExpectedNegamax	[14 7]	1	1	3.966079	14	0.283291
78	Nimby	ExpectedNegamax	[14 7]	1	1	4.018238	18	0.223235
79	Nimby	ExpectedNegamax	[14 7]	1	1	3.985834	20	0.199292
80	Nimby	ExpectedNegamax	[14 7]	1	1	4.014443	18	0.223025
81	Nimby	ExpectedNegamax	[14 7]	1	2	3.857389	15	0.257159
82	Nimby	ExpectedNegamax	[14 7]	1	1	3.965449	18	0.220303
83	Nimby	ExpectedNegamax	[14 7]	1	1	3.960433	18	0.220024
84	Nimby	ExpectedNegamax	[4 3]	1	1	0.107829	20	0.005391
85	Nimby	ExpectedNegamax	[4 3]	1	1	0.113446	20	0.005672
86	Nimby	ExpectedNegamax	[4 3]	1	1	0.112666	20	0.005633
87	Nimby	ExpectedNegamax	[4 3]	1	2	0.109433	19	0.005760
88	Nimby	ExpectedNegamax	[4 3]	1	1	0.109564	20	0.005478
89	Nimby	ExpectedNegamax	[4 3]	1	1	0.106014	18	0.005890
90	Nimby	ExpectedNegamax	[4 3]	1	1	0.110469	20	0.005523
91	Nimby	ExpectedNegamax	[4 3]	1	2	0.109655	19	0.005771
92	Nimby	ExpectedNegamax	[14 7]	2	1	2.893729	17	0.170219
93	Nimby	ExpectedNegamax	[14 7]	2	1	3.054156	19	0.160745
94	Nimby	ExpectedNegamax	[14 7]	2	1	2.795941	19	0.147155
95	Nimby	ExpectedNegamax	[14 7]	2	2	2.801897	18	0.155661
96	Nimby	ExpectedNegamax	[14 7]	2	1	2.800561	19	0.147398
97	Nimby	ExpectedNegamax	[14 7]	2	1	3.094001	19	0.162842
98	Nimby	ExpectedNegamax	[14 7]	2	2	2.816090	18	0.156449
99	Nimby	ExpectedNegamax	[14 7]	2	1	2.916857	19	0.153519
100	Nimby	ExpectedNegamax	[4 3]	2	1	0.097883	19	0.005152
101	Nimby	ExpectedNegamax	[4 3]	2	1	0.101640	19	0.005349
102	Nimby	ExpectedNegamax	[4 3]	2	2	0.102311	18	0.005684
103	Nimby	ExpectedNegamax	[4 3]	2	2	0.102401	20	0.005120
104	Nimby	ExpectedNegamax	[4 3]	2	2	0.100223	18	0.005568
105	Nimby	ExpectedNegamax	[4 3]	2	1	0.093457	17	0.005497
106	Nimby	ExpectedNegamax	[4 3]	2	1	0.094611	17	0.005565
107	Nimby	ExpectedNegamax	[4 3]	2	1	0.101861	19	0.005361

Tabela 1: Zebrane dane umieszczone w pliku .csv zamieszczonego w rozwiązaniu.

## 4.1 Dodatkowe statystyki

- Łączny czas rozgrywek: 81.98712229996454s.
- Średni czas gry: 0.7591400212959679s.
- Średni czas tury: 0.04215094819664955s.
- Średni czas gry Nim: 0.8351441333337183s.
  - z klasycznym algorytmem Negamax: 0.238739025000541s.
  - z algorytmem NoAlphaBetaNegamax: 0.4044398500045645s.
  - z algorytmem ExpectedNegamax: 1.8622535249960492s.
- Średni czas gry Nimby: 0.7496395072912492s.
  - z klasycznym algorytmem Negamax: 0.1640242718749505s.
  - z algorytmem NoAlphaBetaNegamax: 0.3077318812493104s.
  - z algorytmem ExpectedNegamax: 1.7771623687494866s.
- Średni czas tury Nim: 0.04771503806114197s.
  - z klasycznym algorytmem Negamax: 0.015046598389744759s.
  - z algorytmem NoAlphaBetaNegamax: 0.02215057983994484s.
  - z algorytmem ExpectedNegamax: 0.10594794154167175s.
- Średni czas tury Nimby: 0.04145543649792671s.
  - z klasycznym algorytmem Negamax: 0.009587015956640244s.
  - z algorytmem NoAlphaBetaNegamax: 0.015358482487499714s.
  - z algorytmem ExpectedNegamax: 0.09942080825567245s.

## 5 Liczba Wygranych i Przegranych podejść

- Wszystkie gry łącznie: 108.
- Liczba wygranych: 74 (68.52%).
- Liczba przegranych: 34 (31.48%).

W tym dla gier Nim:

- Gracz 1 wygrał: 10 gier typu Nim.
- Gracz 2 wygrał: 2 gry typu Nim.
- Przeprowadzono łącznie 12 gier. Nie było sensu przeprowadzać więcej ponieważ gra jest denterministyczna.

Dla gier Nimby:

- Gracz 1 wygrał: 64 gier typu Nimby.
- Gracz 2 wygrał: 32 gry typu Nim.
- Przeprowadzono łącznie 96 gier.

## 6 Przemyślenia i Wnioski

Wnioski:

- Programowanie dynamicznie znacznie przyspieszyło obliczenia, z uwagi na zmniejszenie powtórnych obliczeń, poprzez zastosowanie tzw.: **Transition Table** - specjalnych tabeli kasz-ujących poszczególne stany gry;
- Czas obliczeń znacznie poprawiło też wyeliminowanie niepotrzebnego kopiowania obiektów, związanego z powrotem do poprzedniego stanu gry.
- Odcinanie za pomocą szacowania alpha / beta poprawiło prawie dwukrotnie czas działania algorytmu AI.
- Nimby jest minimalnie szybszą grą, rozgrywkę prawdopodobnie przyspieszają przypadkowe, niezaplanowane ruchy
- Algorytm ExpectedNegamax jest zdecydowanie wolniejszy od czystego Negamax i wariantu bez odcięcia alpha-beta, znacząco zaniża średnią czasu trwania.

## Literatura

- Biblioteki do kodu:
  - **Numpy**, wersji: 1.26.4
  - **Pandas**, wersji: 2.8.2 - agregacja rezultatów i konwersja do odpowiednich formatów.
  - **EasyAI**, wersji: 2.0.12
- Źródła merytoryczne:
  - [Użyty algorytm Negamax oraz pomoc w implementacji wersji bez Alpha / Beta.](#)
  - [Dokumentacja EasyAI.](#)