Implementacja Monte Carlo Search Tree dla gry Nim

Wybrane zagadnienia sztucznej inteligencji. Zadanie 1.

Maksym Telepchuk

Kwiecień 2020

1 NIM

NIM jest klasycznym przykładem dobrze zanalizowanej od strony matematycznej gry. W tym zadaniu został zaimplementowany trywialny wariant gry. Zasady są takie:

- w szeregu ustawione jest N pionów
- gracz w swojej turze usuwa liczbę pionów od 1 do K
- wygrywa gracz, który usunie ostatni pion z planszy

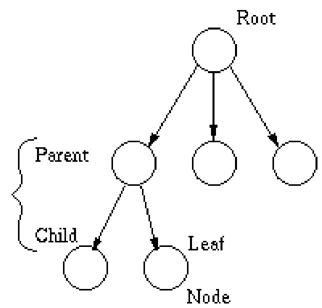
2 MCTS

MCTS (Monte Carlo tree search) to algorytm, który określa najlepszy ruch z zestawu dostępnych ruchów. Algorytm korzysta z formuły **Selekcja** → **Ekspansja** → **Symulacja** → **Aktualizacja**, aby znaleźć ostateczne rozwiązanie. Ten przepływ jest powtarzany wielokrotnie w pętli. W implementacji danego zadnia algorytm będzie ograniczony czasowo.

2.1 Struktura drzewa

MCTS jest algorytmem opierającym się na strukturze danych drzewa. Drzewo składa się z węzłów oraz każdy węzeł jest opisany trzema własnościami:

1. Węzeł rodzica
2. Lista węzłów dzieci
3. Stan węzła



Rysunek 1: Struktura drzewa

Węzeł rodzica pozwala na przemieszczenie się w górę po drzewie, węzły dzieci - w dół. Węzeł początkowy jest nazywany korzeniem i nie ma rodzica. Liściem nazywano węzeł bez dzieci. W tym przypadku liczba dzieci będzie od 1 do K. Stan węzła dla MCTS jest opisywany następująco:

1. Etykieta ruchu
2. Liczba wygranych
3. Liczba rozegranych
4. Stan gry

Etykieta ruchu jest liczbą, określającą ilość zabranych pionków przez przeciwnika w poprzedniej rundzie. Liczba wygranych oraz przegranych są statystykami, potrzebnymi dla rozbudowania drzewa algorytmem MCTS. Liczba wygranych przechowuje wygrane playouty dla gracza, do którego jest przypisany węzeł. Stan gry (w tym przypadku NIMa) jest określony następującymi wartościami:

1. n - aktualna ilość pionów
2. numer gracza - gracz, który w tym momencie podejmuje decyzję ruchu

2.2 Opis algorytmu

Algorytm MCTS przechowuje następujące wartości :

1. Maksymalny czas obliczeń
2. Korzeń drzewa
3. Statystyki obliczeń

W każdej rundzie drzewo MCTS jest budowane od nowa. Algorytmu są przekazywany maksymalny czas obliczeń oraz obecny stan gry. Jedna iteracja pętli Selekcja → Ekspansja → Symulacja → Aktualizacja jest nazywana playoutem. Algorytm wykonuję maksymalną ilość playoutów, które da się rozegrać za podany czas. W trybie analizy, po skończeniu się czasu statystyki drzewa są liczone i zapisywane. W kolejnych podrozdziałach jest opisany proces implementacji playoutu.

2.2.1 Selekcja

Ten proces służy do wybrania węzła w drzewie, który ma najwyższą szansę na wygraną. Wybrany węzeł jest przeszukiwany na podstawie bieżącego stanu drzewa, oraz jest węzłem z nieodwiedzonymi dziećmi lub liściem drzewa. Ponieważ wybrany węzeł ma najwyższą szansę na wygraną - ścieżka przechodząca przez ten węzeł prawdopodobniej dotrze do rozwiązania szybciej niż inna ścieżka w drzewie.

Algorithm 1: Selekcja

```
Result: Wybrany węzeł
node ← root;
while isLeaf(node) == False do
    if hasUnexploredChildren(node) then
        break
    end
    node ← UCT(node) ;
end
return node
```

Jeśli węzeł ma odwiedzone wszystkie dzieci, wtedy kolejny węzeł jest wybierany na podstawie podejścia UCT (**U**pper **C**onfidence bounds applied to **T**rees). To znaczy, że dla każdego dziecka j jest obliczana wartość USB :

$$USB_j = \frac{w_j}{v_j} + C * \sqrt{\frac{ln(v_p)}{v_j}}$$

gdzie

- w_j - ile razy rodzic wygrał wybierając to dziecko
- v_j - ile razy dziecko zostało odwiedzone
- C - współczynnik eksploracji (domyślnie $\sqrt{2}$)
- v_p - ile razy rodzic został odwiedzony

Wybrane zostanie dziecko z najwyższą wartością USB.

2.2.2 Ekspansja

Po wybraniu odpowiedniego węzła jest wykonywana ekspansja. Służy ona do rozszerzania drzewa MCTS w grze poprzez rozwijanie wybranego węzła i tworzenie niezwiedzonych dotychczas węzłów potomnych. W tym przypadku używamy tylko jednego węzła potomnego. Te węzły potomne to przyszłe ruchy, których można użyć w grze. Węzły z warunkiem końcowym (gra się skończyła) nie są dalej rozwijane. Nowy węzeł ma wyzerowaną liczby wygranych i odwiedzonych. Jeśli węzeł ma nieeksplorowane ruchy, z nich jest losowo wybierany jeden ruch oraz tworzono dziecko, z przypisanym stanem gry, która następuje po wybraniu takiego ruchu.

Algorithm 2: Ekspansja

```
Wejście: node: wybrany węzeł
Wyjście: Liść, który jest nowym węzłem lub węzłem z warunkiem końca gry
if isGameOver(node) then
    return node;
end
moves ← getUnexploredMoves(node);
move ← randomItem(moves);
child ← createChild(node,move);
return child
```

2.2.3 Symulacja

Nie wiedząc, który węzeł jest najlepszym pod względem szansy wygrania, rozgrywka powinna być symulowana do końca. Symulacja jest przeprowadzana na kopii stanu gry węzła. Wynik takiej symulacji będzie zatem mieć wpływ na wybór odpowiedniego ruchu. W podstawowej wersji algorytmu, decyzje co do ruchów są podejmowane losowo. Wynikiem takiej symulacji jest gracz, który ją wygrał.

Algorithm 3: Symulacja

```
Wejście: node: węzeł początkowy
Wyjście: winner: gracz, który wygrał symulację
game ← copyGame(node);
while gameOver(game) = False do
    playRandom(game);
    switchPlayer(game);
end
return getWinner(game)
```

2.2.4 Aktualizacja

Na podstawie numeru gracza, który wygrał symulację, wszystkie węzły, które doprowadziły do danego liścia są aktualizowane. Na tej ścieżce w węzłach, które odpowiadają zwycięzcy, liczba wygranych zwiększa się o jeden. Dodatkowo dla każdego węzła na ścieżce liczba rozegranych zwiększa się o jeden.

Algorithm 4: Aktualizcja

```
Wejście: leaf: liść , winner: zwycięzca symulacji
node ← leaf;
while node != null do
    incrementVisitsCount(node);
    if getPlayer(node) = winner then
        incrementWinsCount(node);
    end
    node ← getParent(node);
end
```

2.2.5 Wybór ostateczny

Ostatecznie jest wybierany to dziecko korzenia, które ma największą wartość ilorazu $\frac{\#wygranych}{\#rozegranych}$ dla gracza, przypisanego do korzenia.

2.3 Proponowane polepszenia

2.3.1 Nie losowa strategia

Warto sprawdzić, czy da się polepszyć działanie algorytmu wprowadzając małe zmiany. Pierwszą taką zmianą jest strategia wybierania kolejnego ruchu. Jeśli algorytm, zamiast losowej symulacji, będzie pozostawiał liczbę pionów, która jest podzielna przez $K + 1$, wtedy szansa na wygranie może się zwiększyć. Przeciwnik algorytmu przy $K + 1$ pionach nie jest w stanie wygrać, ponieważ algorytm zabierze pozostałe piony i wygra grę.

Dotrzymywanie liczby pionów podzielnej przez $K + 1$ podobno zwiększa szansę na pozytywny koniec gry. Gdy liczba pionów już jest podzielna przez $K+1$, algorytm usuwa losową ilość pionów. Zachowanie przeciwnika pozostaje się losowym.

2.3.2 Przemieszczenie się po drzewie

W każdej rundzie drzewo MCTS jest budowane od nowa. Przemieszczanie korzenia w dół po drzewie na podstawie wybranych ruchów może pozwolić na korzystanie z już obliczonych archiwalnych danych oraz w teorii może polepszyć wyniki algorytmu.

3 Analiza

Dla analizy został zaimplementowany automatyczny symulator, który prowadzi gry. Testowano 4 typy botów:

1. Bot losowy
2. Losowy MCTS
3. Strategiczny MCTS
4. MCTS z archiwalnymi danymi

Początkowa liczba pionów (od 20 do 50) oraz pierwszy gracz są ustawiane losowo. K jest ustawiona na 3, zgodnie z treścią zadania. Z każdej gry są zbierane i zapisywane następujące statystyki:

1. n : liczba pionów w czasie podejmowania ruchu
2. popularity : ile razy wystąpiła potrzeba podjęcia decyzji dla n
3. avg. convincity : średni współczynnik pewności dla wybranego dziecka
4. avg. playouts : średnia ilość playoutów
5. avg. path : średnia długość ścieżek od korzenia do liści
6. med. path : mediana długości ścieżek od korzenia do liści
7. max. path : maksymalna długość ścieżki od korzenia do liści
8. games win : wygrane gry dla danego bota

3.1 Bot losowy vs. Losowy MCTS

Liczba gier : 300
Czas myślenia : 10 ms
Liczba wygranych dla bota losowego : 297
Liczba wygranych dla losowego MCTS : 3
Wszystkie statystyki znajdują się w Tablice 1.
Najbardziej istotne wykresy są przestawione w rysunku 2.

3.2 Losowy MCTS vs. Strategiczny MCTS

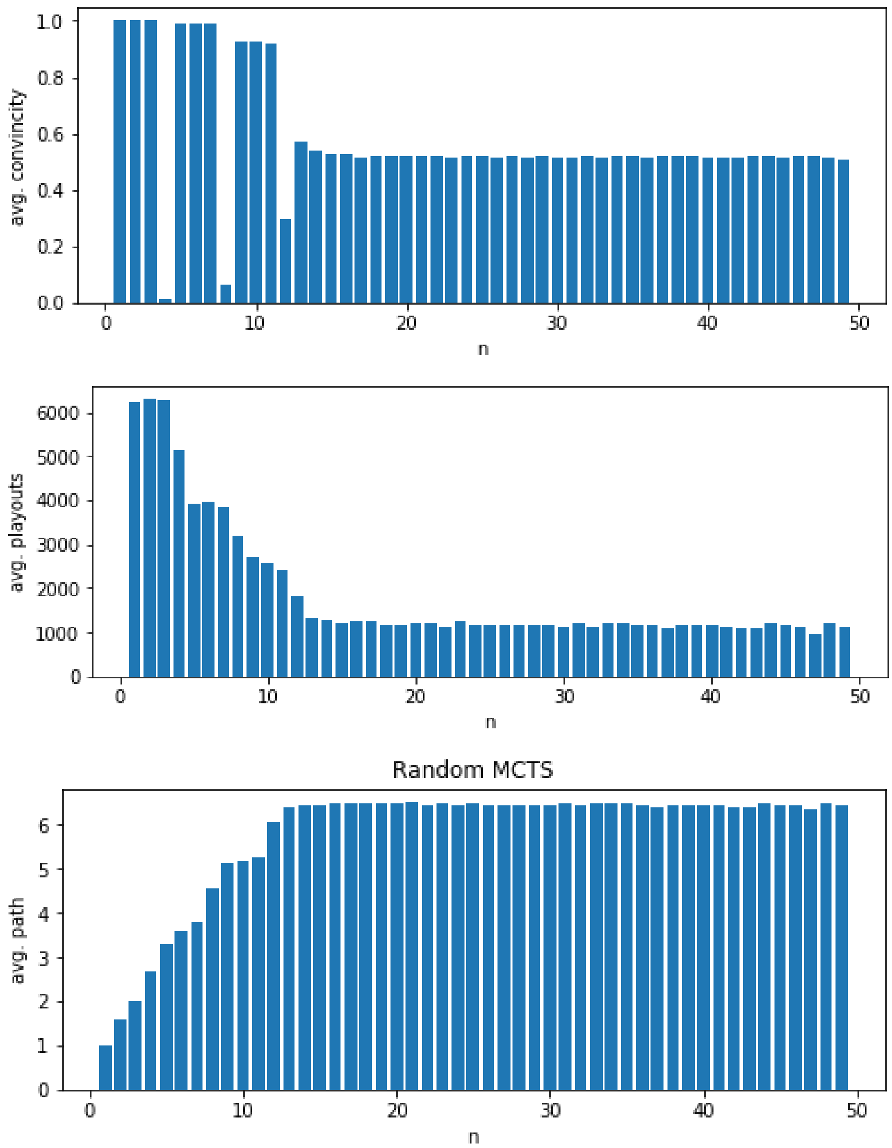
Liczba gier : 300
Czas myślenia : 10 ms
Liczba wygranych dla strategicznego MCTS : 143
Liczba wygranych dla losowego MCTS : 157
Wszystkie statystyki dla strategicznego MCTS znajdują się w Tablice 2.
Najbardziej istotne wykresy są przestawione w rysunku 3.

3.3 Losowy MCTS vs. Archiwalny MCTS

Liczba gier : 300
Czas myślenia : 10 ms
Liczba wygranych dla losowego MCTS : 217
Liczba wygranych dla archiwalnego MCTS : 83
Wszystkie statystyki dla strategicznego MCTS znajdują się w Tablice 3.

n	popularity	avg. convincity	avg. playouts	avg. path	med. path	max. path
1	97	1.0000	6745.9072	1.0000	1	1
2	95	1.0000	6749.7789	1.6000	2	2
3	105	1.0000	6620.1810	1.9986	2	3
4	6	0.0139	5403.6667	2.6667	3	4
5	102	0.9903	4253.7549	3.2949	3	5
6	91	0.9903	4238.2308	3.5820	4	5
7	99	0.9899	4141.1818	3.7998	4	5
8	24	0.0603	3291.5833	4.5592	5	6
9	80	0.9343	2865.2875	5.1409	5	7
10	89	0.9280	2798.4382	5.1968	5	7
11	96	0.9298	2729.3229	5.2668	5	7
12	79	0.2770	1939.6582	6.0946	6	8
13	68	0.5826	1480.0147	6.4599	6	9
14	63	0.5367	1390.0159	6.4834	6	9
15	85	0.5306	1304.6000	6.4736	6	9
16	71	0.5192	1306.8873	6.5079	7	9
17	76	0.5156	1251.9737	6.4903	6	9
18	80	0.5175	1258.8500	6.5094	7	9
19	71	0.5209	1247.0141	6.5090	6	9
20	71	0.5182	1222.8592	6.4926	6	9
21	69	0.5145	1208.4928	6.4744	6	9
22	67	0.5174	1236.2388	6.5075	6	9
23	80	0.5184	1226.6875	6.4926	6	9
24	73	0.5157	1229.5068	6.5003	6	9
25	60	0.5169	1248.5333	6.5028	6	9
26	58	0.5167	1203.2241	6.4761	6	9
27	61	0.5174	1178.6557	6.4668	6	9
28	62	0.5158	1215.7097	6.4801	6	9
29	46	0.5209	1174.0217	6.4691	6	9
30	59	0.5189	1202.4915	6.4720	6	9
31	45	0.5180	1200.6889	6.5021	7	9
32	45	0.5170	1186.1111	6.4605	6	9
33	38	0.5164	1165.2632	6.4373	6	9
34	53	0.5174	1193.8679	6.4567	6	9
35	33	0.5123	1179.2121	6.4641	6	9
36	30	0.5130	1187.9667	6.4585	6	9
37	35	0.5171	1156.0857	6.4451	6	9
38	38	0.5193	1201.5263	6.4670	6	9
39	26	0.5103	1148.6923	6.4330	6	9
40	33	0.5207	1141.2727	6.4272	6	9
41	18	0.5185	1205.1667	6.4697	6	9
42	26	0.5204	1138.2308	6.4171	6	9
43	24	0.5166	1138.1667	6.4212	6	9
44	8	0.5127	1129.0000	6.3957	6	8
45	19	0.5174	1144.9474	6.4151	6	9
46	11	0.5181	1235.4545	6.4940	6	9
47	9	0.5095	1150.8889	6.4400	6	9
48	7	0.5130	1156.1429	6.4218	6	8
49	3	0.5021	1187.0000	6.4272	6	8

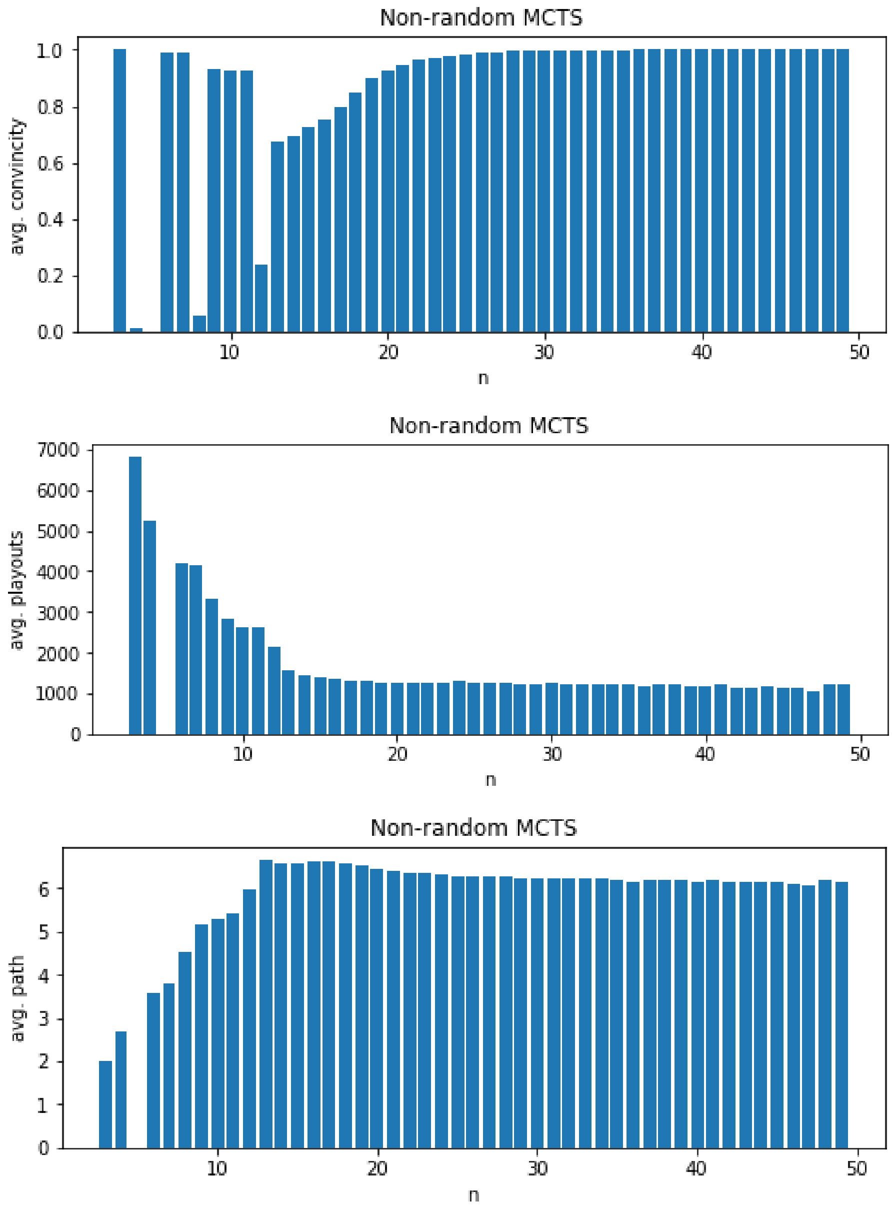
Tablica 1: Statystyki losowego MCTS



Rysunek 2: Losowy MCTS

n	popularity	avg. convincity	avg. playouts	avg. path	med. path	max. path
3	143	1.0000	6732.1608	2.0000	2	3
4	157	0.0152	5153.0637	2.6662	3	4
6	29	0.9904	4144.6552	3.5749	4	5
7	114	0.9904	4130.2895	3.7756	4	5
8	157	0.0589	3269.7261	4.5068	5	6
9	5	0.9367	2863.4000	5.1437	5	7
10	44	0.9250	2561.5000	5.2866	5	7
11	94	0.9258	2591.1170	5.4001	5	7
12	155	0.2446	2068.2129	5.9543	6	7
13	140	0.6806	1514.2071	6.6129	7	9
14	87	0.6991	1414.5977	6.5637	7	9
15	68	0.7287	1386.0588	6.5684	7	9
16	114	0.7564	1308.4912	6.5877	7	9
17	95	0.7996	1279.1684	6.6025	7	10
18	96	0.8532	1289.9375	6.5866	7	9
19	80	0.8995	1256.9750	6.4890	6	9
20	89	0.9226	1298.3371	6.4410	6	8
21	100	0.9454	1255.6700	6.3833	6	8
22	88	0.9594	1281.3750	6.3714	6	8
23	84	0.9713	1292.0357	6.3469	6	8
24	61	0.9778	1276.3115	6.3176	6	7
25	83	0.9852	1280.4578	6.3055	6	7
26	67	0.9884	1310.6119	6.3240	6	7
27	68	0.9923	1272.4706	6.2889	6	7
28	54	0.9948	1242.7222	6.2622	6	7
29	57	0.9965	1272.5263	6.2822	6	7
30	55	0.9979	1271.2364	6.2760	6	7
31	43	0.9984	1256.1395	6.2748	6	7
32	41	0.9988	1288.1951	6.2830	6	7
33	47	0.9996	1283.2553	6.2885	6	7
34	43	0.9999	1230.5116	6.2300	6	7
35	37	0.9997	1225.8378	6.2267	6	7
36	34	0.9999	1251.9706	6.2572	6	7
37	32	1.0000	1233.3125	6.2185	6	7
38	30	1.0000	1287.4667	6.2696	6	7
39	29	1.0000	1220.9310	6.2496	6	7
40	24	1.0000	1236.2917	6.2298	6	7
41	23	1.0000	1249.4348	6.2428	6	7
42	18	1.0000	1221.2778	6.2402	6	7
43	24	1.0000	1259.8750	6.2327	6	7
44	13	1.0000	1223.8462	6.2414	6	7
45	11	1.0000	1216.7273	6.1968	6	7
46	18	1.0000	1236.4444	6.2302	6	7
47	6	1.0000	1220.8333	6.2099	6	7
48	9	1.0000	1266.7778	6.2420	6	7
49	6	1.0000	1228.1667	6.2360	6	7

Tablica 2: Statystyki strategicznego MCTS



Rysunek 3: Strategiczny MCTS

n	popularity	avg. convincity	avg. playouts	avg. path	med. path	max. path
3	83	1.0000	6663.0000	2.0000	2	3
4	67	0.0125	5066.2985	2.6667	3	4
5	45	0.9920	4358.6444	3.2947	3	5
6	79	0.9920	4238.2911	3.5938	4	5
7	109	0.9922	4365.8991	3.8199	4	5
8	68	0.0505	3494.9412	4.5677	5	6
9	44	0.9373	2903.0909	5.1418	5	7
10	105	0.9399	3017.3048	5.2203	5	7
11	83	0.9396	2999.0723	5.2734	5	7
12	83	0.2403	2151.8675	6.1327	6	8
13	74	0.6453	1589.1622	6.5544	7	9
14	87	0.5579	1364.8736	6.5714	7	9
15	83	0.5387	1289.9398	6.5615	7	9
16	63	0.5253	1280.3175	6.6147	7	9
17	77	0.5144	1255.9610	6.6342	7	9
18	75	0.5155	1248.3200	6.6290	7	9
19	77	0.5189	1267.8961	6.6377	7	9
20	72	0.5210	1190.5417	6.5803	7	9
21	78	0.5185	1243.1795	6.6138	7	9
22	73	0.5167	1244.5890	6.6166	7	9
23	71	0.5170	1200.2958	6.5949	7	9
24	66	0.5158	1206.0000	6.5822	7	9
25	66	0.5174	1223.4697	6.6092	7	9
26	67	0.5169	1227.4478	6.5934	7	9
27	57	0.5207	1200.1930	6.5855	7	9
28	59	0.5169	1165.9322	6.5714	7	9
29	57	0.5185	1196.6316	6.5750	7	9
30	59	0.5199	1195.7119	6.5911	7	9
31	55	0.5138	1231.4727	6.5859	7	9
32	39	0.5177	1182.3333	6.5598	7	9
33	57	0.5178	1179.1579	6.5612	7	9
34	39	0.5189	1152.7692	6.5403	7	9
35	50	0.5159	1139.5000	6.5328	7	9
36	30	0.5234	1184.9333	6.5484	7	9
37	39	0.5174	1176.0513	6.5311	7	9
38	36	0.5179	1172.7500	6.5406	7	9
39	31	0.5156	1209.3548	6.5364	7	9
40	25	0.5232	1119.4000	6.5099	7	9
41	25	0.5186	1085.9600	6.4680	6	9
42	23	0.5200	1166.2609	6.5025	6	9
43	19	0.5217	1141.7895	6.4912	6	9
44	21	0.5129	1073.9524	6.4366	6	9
45	11	0.5260	1054.1818	6.4237	6	9
46	14	0.5163	1152.7143	6.4794	6	9
47	7	0.5245	1095.4286	6.3995	6	8
48	13	0.5104	973.0000	6.3562	6	9
49	8	0.5212	1134.0000	6.4131	6	8

Tablica 3: Archiwalny MCTS

3.4 Losowy MCTS vs. Zachłannej strategii

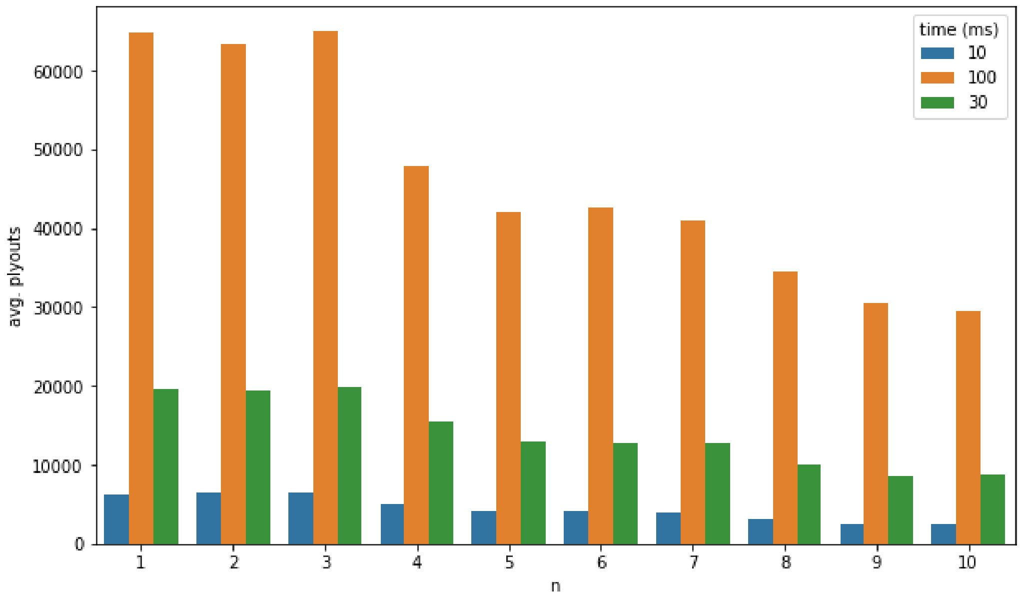
Liczba gier : 300
Czas myślenia : 100 ms
Liczba wygranych dla losowego MCTS : 2
Liczba wygranych dla archiwalnego MCTS : 298

Podejście zachłannego agenta bazuje się na strategii wybierania reszty z dzielenia n przez K + 1. Jeśli ta reszta jest 0, to kolejny ruch jest wybierany jako 1.

Jak widać, zwykłe podejście heurestyczne jest bardzo skuteczne, więc implementacja MCTS w danej grze nie jest najlepszym pomysłem.

3.5 Zależność od czasu

Z wyniku poprzednich rozdziałów widać, że liczba playoutów zaczyna istotnie rosnąć w okolicach n = 10. Żeby zbadać zależność pomiędzy liczbą playoutów a czasem, zostało symulowano 300 gier z początkową N = 10 dla czasów 10 ms, 30 ms oraz 100 ms. Na rys. 4 widać, że zależność liczby playoutów od czasu pod koniec gry jest liniowa.



Rysunek 4: Liczba playoutów dla różnych czasów

4 Wnioski

Algorytm MCTS jest algorytmem, który przeprowadza tysiące gier, żeby zrobić jeden ruch. Algorytm wykrywa strategie, które istnieją w grze i wygrywa strategie wymyślone przez człowieka.

Analiza proponowanych polepszeń algorytmu wykazała, że narzucanie na pierwszy rzut oka sensownego podejścia nie zawsze działa lepiej od podejścia losowego błądzenia. Zmieniony algorytm jest bardzo pewny swoich ruchów na początku gry, jednak w walce przeciwko nie losowego algorytmu wykazał niską pewność swoich ruchów bliżej do końca gry. W przeciwieństwu do tego, przy losowej strategii algorytm stawał się coraz bardziej pewny swoich ruchów.

Przechowywanie danych archiwalnych również okazało się nie przydatne w zmieniającym się środowisku.

Z wykresów też wynika, że przy każdym analizowanym podejściu liczba playoutów większość gry jest mniej więcej taka sam i zaczyna rosnąć pod koniec gry. Prawdopodobnie to jest związane z tym, że bliżej końca gry każdy węzeł w drzewie jest zbadany (o czym swiadczą wykresy średnich ścieżek, które są podobne w każdym podejściu).

Przy losowej strategii maksymalna głębokość węzłów jest wyższa, co świadczy o tym, że losowa strategia rozpatruje więcej przypadków, niż przy narzuconej strategii.

Na koniec warto wspomnieć, że zwykłe heurestyczne jest wstanie dać skuteczniejsze wyniki od MCTS, więc dany wariant gry jest zbyt prosty dla tego, żeby stosować w nim MCTS.

5 Nim z monetą

Żeby implementacja MCTS nie wyglądała tak beznadziejnie w porównaniu do prostej strategii heurystycznej, zaproponowano i zbadano następującą zmianę zasad gry w NIM.

Po dowolnym ruchu każdy gracz rzuca monetą. Jeśli wypadł orzeł - z planszy dodatkowo jest zabierany jeden pion. Jeśli w konsekwencji rzutu monetą został zabrany ostatni pion z planszy, wygrywa ten gracz, który robił ruch ostatnim (czyli ten, który rzucał monetą).

Przy takich zasadach, gdy gracz ma do czynienia z K + 1 pionami, sytuacja nie jest przegrana - można zabrać K pionów i mieć nadzieję, że wypadnie orzeł, co usunie ostatni pion z planszy. Podejście zachłanne tutaj będzie przegrywało w 50% przypadkach. Ale czy można uzyskać lepsze wyniki?

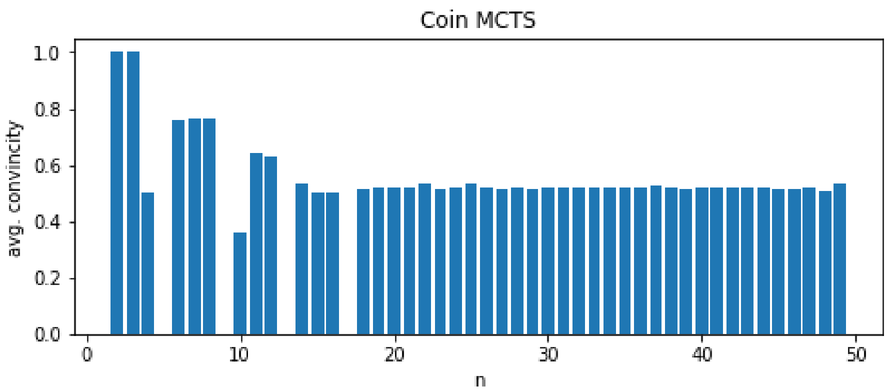
5.1 Zmiany do MCTS

Wartości w węzłach MCTS pozostają się bez zmian. Zmienia się tylko sposób symulowania playoutu. Mianowicie, przy każdym przejściu w dół po drzewie MCTS rzuca monetą i zlicza ilość orłów. Jeśli w dowolnym momencie playoutu liczba pionów w aktualnie rozpatrywanym węźle jest mniejsza od liczby zliczonych dotychczas orłów, to playout jest przerywany, oraz aktualizacja drzewa jest wykonywana od węzła, w którym gra została przerwana.

Daje to możliwość MCTS nauczyć się uwzględniać takie losowe wydarzenie, jak rzut monetą.

5.2 MCTS vs. Zachłannej strategii w Nimie z monetą

Liczba gier : 300
Czas myślenia : 10 ms
Liczba wygranych dla MCTS : 258
Liczba wygranych dla strategii zachłannej : 42
Wszystkie statystyki dla strategicznego MCTS znajdują się w Tablice 4.
Wykresy pewności jest przedstawiony w rysunku 5.



Rysunek 5: Pewność ruchów dla MCTS z monetą

6 Ostateczne wyniki

Wyniki badania wykazują, że MCTS w większości przypadków nie jest pewniejszy swoich ruchów w porównaniu do NIMa bez rzutu monetą (**Uwaga**: algorytm nawet nie miał do czynienia z sytuacją 5 i 9 pionów na planszy). Jednak algorytm nie wpada w beznadziejne sytuacje jak to (4 i 8 pionów, gdy algorytm był bardzo niepewny swoich ruchów) i wykazuje lepszą tendencję do wygrywania w grze z prostym podejściem heurestycznym

Między innym, sytuacja dla 4 pionów wydaję się MCTS w 50% wygrana, co wskazuje na to, że w tym przypadku MCTS wybiera 3 piony i czeka na orła, ponieważ to jedyna szansa wygrać grę.

n	popularity	avg. convincity	avg. playouts	avg. path	med. path	max. path
2	79	1.0000	13222.8481	1.6000	2	2
3	135	1.0000	13401.4074	1.9855	2	3
4	86	0.5007	10205.4186	2.5981	3	4
6	7	0.7635	6145.7143	3.5720	4	5
7	142	0.7616	5859.9930	3.9387	4	5
8	136	0.7631	5812.9559	4.2398	4	6
10	32	0.3609	3622.2188	5.2062	5	7
11	117	0.6439	2898.9060	5.6244	6	8
12	91	0.6259	2380.4505	5.8013	6	8
14	24	0.5292	1517.4583	6.1194	6	8
15	135	0.5028	1386.0370	6.2441	6	8
16	95	0.5007	1260.7263	6.3033	6	8
18	42	0.5224	1175.3810	6.3883	6	9
19	128	0.5185	1129.9219	6.3874	6	9
20	95	0.5179	1111.2421	6.3910	6	9
21	8	0.5130	981.0000	6.3348	6	8
22	29	0.5235	1074.1724	6.3887	6	9
23	107	0.5186	1078.1308	6.3899	6	9
24	98	0.5175	1091.7857	6.3883	6	9
25	1	0.5170	685.0000	5.9295	6	7
26	36	0.5160	1065.2222	6.3734	6	9
27	99	0.5178	1108.4040	6.3983	6	9
28	69	0.5174	1081.5797	6.3818	6	9
29	7	0.5166	1086.8571	6.3771	6	9
30	27	0.5248	1040.9259	6.3607	6	9
31	76	0.5165	1111.8421	6.3978	6	9
32	62	0.5165	1076.1452	6.3805	6	9
33	6	0.5165	693.5000	6.0281	6	9
34	29	0.5224	998.3448	6.3156	6	8
35	54	0.5148	1089.4259	6.3717	6	9
36	50	0.5167	1052.3000	6.3622	6	8
37	4	0.5285	1134.0000	6.4037	6	8
38	10	0.5169	1152.9000	6.4265	6	8
39	43	0.5152	1027.6512	6.3341	6	9
40	34	0.5167	1115.0588	6.4025	6	9
41	4	0.5148	1127.7500	6.4182	6	8
42	12	0.5165	1112.5833	6.3794	6	8
43	31	0.5179	976.8710	6.3072	6	8
44	17	0.5216	1077.0000	6.3870	6	8
45	5	0.5181	1044.4000	6.3400	6	8
46	7	0.5269	1094.0000	6.3865	6	8
47	13	0.5076	1063.0769	6.3469	6	8
48	8	0.5163	957.2500	6.2751	6	8
49	7	0.5181	849.8571	6.1983	6	8

Tablica 4: Statystyki MCTS w Nimie z monetą