

Projektowanie Algorytmów i Metody Sztucznej Inteligencji

Projekt 3

Kółko i krzyżyk

Data oddania sprawozdania	05.06.2019
Imię i nazwisko	Kinga Tokarska 241621
Nr indeksu	241621
Termin zajęć	Środa, 11.15-13.00
Prowadzący kurs	Dr inż. Łukasz Jeleń
Kod kursu	E02-47f

1. Wstęp teoretyczny

Teoria gier i konstruowane na jej podstawie programy stanowią jeden z głównych wyznaczników postępu sztucznej inteligencji. Teoria gier to gałąź matematyki zajmująca się zachowaniem w sytuacjach konfliktowych. Grą nazywamy dowolną sytuację konfliktową prowadzoną przez graczy zgodnie z ustalonymi zasadami. Każdy z graczy ma do wyboru pewną liczbę możliwych strategii, określających dokładnie sposób jego postępowania w rozgrywce, którego celem jest osiągnięcie określonego rezultatu. Wynik gry jest jednoznacznie określony przez kombinację strategii wybranych przez graczy.

Gry można podzielić na kilka podtypów. Najlepiej rozwiniętą i dającą najbardziej konkretne wyniki jest teoria gier dwuosobowych o sumie stałej. Grą o sumie stałej nazywamy grę, w której zysk jednego gracza oznacza stratę drugiego. Szczególnym przypadkiem gry o sumie stałej jest gra o sumie zerowej. Suma zerowa oznacza, że zwycięstwo jednego gracza jest równoznaczne z porażką drugiego. Suma nagród i kar jest stała i symetryczna, równa zero.

Przykładem gry skończonej o sumie zerowej przeznaczonej dla dwóch graczy, jest popularna strategiczna gra umysłowa „Kółko i krzyżyk”. Zaliczamy ją do gier skończonych, z pełną informacją, to znaczy obaj gracze mają dokładną wiedzę o sytuacji w grze i istnieje gwarancja, że każda rozgrywka ulegnie zakończeniu. Jest to jednocześnie pierwsza logiczna gra, która została opanowana przez maszynę. Komputer pokonał człowieka w grze w „Kółko i krzyżyk” już w latach pięćdziesiątych. [1]

2. Opis zastosowanych algorytmów sztucznej inteligencji

Twierdzenie o Minimaksie mówi, że dla każdej skończonej gry dwuosobowej o sumie zero, istnieje przynajmniej jedna optymalna strategia mieszana. Tym samym istnieje wartość gry V , taka że gracz pierwszy, stosując optymalną strategię gwarantuje sobie oczekiwaną wypłatę nie gorszą niż V , natomiast jego przeciwnik, gracz drugi, stosując optymalną strategię gwarantuje sobie wypłatę nie gorszą niż $-V$.

Algorytm Minimax to klasyczny algorytm stosowanym w prostych grach logicznych do wyznaczania optymalnego ruchu. Jest algorytmem rekurencyjnym, który wywołuje sam siebie do analizy kolejnych ruchów w grze. Oparty jest na funkcji, oceniającej wartość stanu gry w dowolnym momencie. Zakłada on istnienie dwóch graczy. Gracz pierwszy, określany dalej jako gracz dąży do zmaksymalizowania stanu, a gracz drugi, określany dalej jako przeciwnik dąży do zminimalizowania stanu. Oznacza to, że gracz stara się wykonać ruch najkorzystniejszy dla siebie i najmniej korzystny dla przeciwnika, i odwrotnie - przeciwnik stara się wykonać ruch najkorzystniejszy dla siebie i najmniej korzystny dla gracza.

Funkcja zostaje wykorzystana do obliczenia drzewa wszystkich możliwych stanów w grze do pewnej głębokości, ograniczonej przez moc obliczeniową. Drzewo gry jest spójnym, acyklicznym grafem. Węzły drzewa reprezentują możliwe stany gry na planszy. Korzeń drzewa jest stanem początkowym gry, czyli stanem w którym poszukujemy najlepszego ruchu dla gracza. Krawędzie drzewa reprezentują ruchy gracza i przeciwnika, prowadzące z jednego stanu do drugiego. Gałęzie drzewa wychodzące z każdego węzła reprezentują wszystkie możliwe ruchy odpowiednio gracza i przeciwnika. Każda z tych gałęzi prowadzi do węzła potomnego związanego z kolejną sytuacją na planszy, osiąganą po wykonaniu odpowiedniego ruchu, w której ruch będzie wykonywać drugi z graczy.

Złożoność klasycznej gry w „Kółko i krzyżyk” jest równa 26830. Drzewo tej gry jest więc łatwe do przeszukania. W pełnym drzewie liście, czyli gałęzie terminalne reprezentują sytuacje, w których partia gry jest rozstrzygnięta. Mając kompletne drzewo gry, można w pełni rozwiązać tę grę, to znaczy znaleźć sekwencję ruchów, które zawsze gwarantują zwycięstwo lub remis. Po rozszerzeniu klasycznej wersji gry, to znaczy zwiększeniu rozmiaru planszy lub modyfikacji zasad rozgrywki, zastosowanie zasady Minimax wprost jest niemożliwe. Kompletne drzewo takiej gry byłoby zbyt duże. Konieczne jest więc przejrzanie okrojonej wersji drzewa gry, stosując funkcje oceniające wartościujące jego liście w sposób heurystyczny.

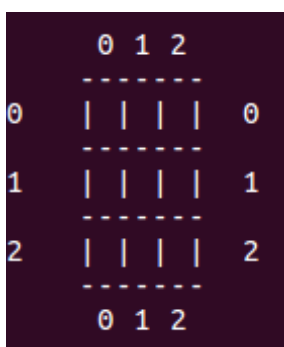
Funkcja oceniająca przyporządkowuje danej pozycji wartość z pewnego zbioru, w przypadku algorytmu Minimax jest to wartość liczbowa. Idealną funkcją oceniającą jest więc funkcja, która zawsze zwraca jako wynik dokładną wartość minimaksowej pozycji. Skonstruowanie takiej funkcji jest na ogół niemożliwe. Konieczne jest skonstruowanie funkcji oceniających opartych na przesłankach heurystycznych. Węzłom przypisuje się liczbową ocenę ich użyteczności z punktu widzenia bieżącego gracza w następujący sposób: ocena dodatnia ∞ w przypadku wygranej gracza, ocena ujemna $-\infty$ w przypadku wygranej przeciwnika i ocena neutralna 0 w przypadku remisu. W przypadku braku rozstrzygnięcia gry na aktualnej planszy, algorytm oblicza wartość planszy w oparciu o ilość elementów w rzędzie każdego z graczy. Im większa jest wartość węzła, tym korzystniejsza jest sytuacja z punktu widzenia przeciwnika (komputera), a im mniejsza wartość węzła tym sytuacja jest korzystniejsza z punktu widzenia użytkownika. Po dokonaniu w opisany sposób oceny wszystkich węzłów terminalnych drzewa gry, przystępuje się do oceny węzłów wewnętrznych. Na poziomie gracza węzeł otrzymuje ocenę równą maksimum ocen jego węzłów potomnych, a na poziomie przeciwnika węzeł otrzymuje ocenę równą minimum ocen jego węzłów potomnych.

Algorytm Minimax jest zadowalającym rozwiązaniem przy klasycznej wersji gry w kółko i krzyżyk (plansza o rozmiarze 3x3, wygrana po utworzeniu linii złożonej z trzech symboli). W rozbudowanej wersji gry (plansza o rozmiarze większym niż 3x3, wygrana po utworzeniu linii złożonej z więcej niż trzech symboli) opisany algorytm jest porównywalny z przeciętnym ludzkim graczem, ale ustępuje bardziej doświadczonym i błyskotliwym graczom. Aby jednak wykazać dobry poziom gry potrzebny jest algorytm w bardziej racjonalny sposób wykorzystujący dostępną moc obliczeniową w celu możliwie głębokiego zbadania drzewa gry tam gdzie może to wpłynąć na zmianę wybieranego ruchu, kosztem rezygnacji z rozbudowy i analizy drzewa gry tam, gdzie jej wynik i tak nie wpłynie na wybierany ruch. Taką zrationalizowaną alokację mocy obliczeniowej realizuje algorytm cięć alfa-beta, oparty na obciążonym algorytmie mini-maks, lecz wzbogacający go o kryteria umożliwiające bezpieczne pominięcie w analizie fragmentów drzewa gry. Cięcie alfa polega na ocenianiu węzła przez maksymalizację ocen węzłów potomnych i możliwości zakończenia wyznaczania oceny węzła potomnego natychmiast po stwierdzeniu, że musi być ona niższa niż dotychczasowe maksimum alfa. Cięcie beta polega na ocenianiu węzła przez minimalizację ocen węzłów potomnych i możliwości zakończenia wyznaczania oceny węzła potomnego natychmiast po stwierdzeniu, że musi być ona wyższa niż dotychczasowe minimum beta. Ze względu na wyróżnienie dwóch rodzajów cięć, odpowiadających poziomom gracza i przeciwnika w drzewie gry, algorytm sformułowany jest za pomocą dwóch procedur z pośrednią rekurencją. Procedura alfa stosowana jest dla węzłów poziomu gracza (w tym dla węzła początkowego, w którym poszukujemy ruchu do wykonania), zaś procedura beta dla poziomu przeciwnika. [2]

3. Zasady gry

1) Gra „Kółko i krzyżyk” rozgrywana jest na kwadratowej planszy – o takiej samej ilości wierszy i kolumn. Domyślnie plansza ma wymiary 3x3, ale istnieje możliwość zmiany wielkości planszy. Wiersze i kolumny zostały ponumerowane liczbami z zakresu 0-3.

2) Początkowy stan gry to pusta plansza podzielona na pola, na których gracze będą stawiać swoje symbole. Liczba pól równa jest iloczynowi wierszy i kolumn. Każde pole planszy można opisać współrzędnymi (a,b), gdzie a to numer wiersza, a b to numer kolumny.



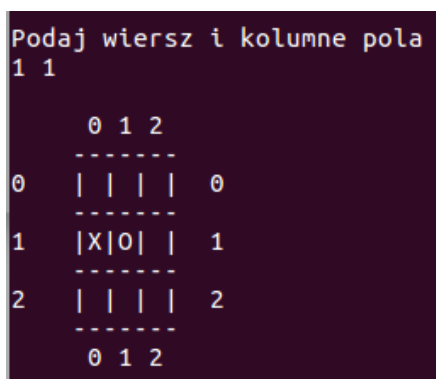
	0	1	2	
0				0
1				1
2				2
	0	1	2	

Rys. 1. Plansza przed rozpoczęciem rozgrywki

3) Rozgrywka toczy się pomiędzy dwoma graczami. W trybie gry użytkownik-komputer grę rozpoczyna użytkownik.

4) Program prosi o podanie współrzędnych, czyli numeru wiersza i kolumny. Użytkownik wybiera pole, na którym ma zostać postawione kółko (symbol o), podając jego współrzędne.

5) Następnie automatycznie ruch wykonuje przeciwnik, czyli komputer, stawiając na planszy kółko (symbol o). Program wyświetla aktualny stan planszy, to znaczy planszę z kółkiem (symbol o) postawionym przez użytkownika i krzyżykiem (symbol x) postawionym przez komputer.



Podaj wiersz i kolumnę pola				
1 1				
	0	1	2	
0				0
1	x o			1
2				2
	0	1	2	

Rys. 2. Plansza po pierwszej turze (ruch użytkownika i komputera)

7) Gracze – użytkownik i komputer wykonują ruchy naprzemiennie, stawiając po jednym, odpowiednim dla siebie symbolu (krzyżyku lub kółku).

```

Podaj wiersz i kolumnę pola
0 0

      0 1 2
      -----
0    |0| | | 0
      -----
1    |X|0| | 1
      -----
2    | | |X| 2
      -----
      0 1 2
  
```

Rys. 3. Plansza po kolejnej turze (ruch użytkownika i komputera)

8) Nie można postawić symbolu na polu zajęтым przez przeciwnika. Pole zajęte przez jednego gracza, nie zmienia swojego właściciela do końca rozgrywki.

```

      0 1 2
      -----
0    | | | | 0
      -----
1    |X|0| | 1
      -----
2    | | | | 2
      -----
      0 1 2

Podaj wiersz i kolumnę pola
1 0
Podaj wiersz i kolumnę pola
  
```

Rys. 4. Nieprawidłowy sposób wykonywania ruchu

9) Celem gry jest postawienie przez gracza linii odpowiedniej dla wielkości planszy ilości odpowiednich dla niego symboli (krzyżyków lub kółek). Symbole mogą być ustawione w linii pionowej, poziomej lub ukośnej. Domyślnie liczba wymaganych symboli wynosi 3, ale istnieje możliwość jej modyfikacji.

```

      0 1 2
      -----
0    |0|0|0| 0
      -----
1    |X|0| | 1
      -----
2    |X| |X| 2
      -----
      0 1 2

GRACZ WYGRAL
  
```

Rys. 5. Rozgrywka zakończona wygraną użytkownika

10) Gdy wszystkie pola na planszy zostaną wypełnione i żaden gracz nie osiągnie celu gry, rozgrywka kończy się remisem. [3]

	0	1	2	

0	O X			0

1	X X O			1

2	O O X			2

	0	1	2	

Rys. 6. Rozgrywka zakończona remisem

4. Wyniki rozgrywek

Zostało przeprowadzone doświadczenie polegające na rozegraniu dziesięciu rozgrywek przy trzech różnych rozmiarach planszy oraz ilościach symboli zapewniających wygraną. Wyniki eksperymentu (zwycięzcy poszczególnych rozgrywek) przedstawione zostały w poniższej tabeli.

Numer rozgrywki	Rozmiar planszy (ilość symboli w linii zapewniająca wygraną)		
	3x3 (3)	5x5 (4)	7x7 (5)
1	remis	remis	remis
2	komputer	remis	użytkownik
3	remis	użytkownik	użytkownik
4	remis	remis	remis
5	komputer	komputer	komputer
6	remis	użytkownik	użytkownik
7	remis	remis	remis
8	remis	komputer	użytkownik
9	komputer	użytkownik	użytkownik
10	komputer	komputer	użytkownik

Tab. 1. Wyniki rozgrywek

5. Wnioski

Za podstawową umiejętność programu należy uznać wykrywanie możliwych ruchów wygrywających i stosowaniu ich w swojej strategii lub blokowaniu ich w strategii przeciwnika. Wszystkie ruchy powinny prowadzić do połączenia symboli i utworzenia jak największej ilości możliwych linii-zagrożeń, oraz rozbijania grup symboli przeciwnika.

Zastosowany w programie algorytm Minimax okazuje się niezwykle skuteczny. Dzięki jego zastosowaniu, w klasycznej wersji gry „Kółko i krzyżyk” komputer staje się przeciwnikiem nie do pokonania. Dla rozszerzonej wersji gry oraz bardziej skomplikowanych gier takich jak warcaby czy szachy może on jednak stawać się zbyt czasochłonny lub pamięciożerny.

Przy małym rozmiarze planszy możliwe jest zastosowanie algorytmu analizującego dużą ilość ruchów. Dzięki temu początkujący użytkownik ma niewielkie szanse na zwycięstwo z komputerem. Z kolei przy bardziej doświadczonych graczach, gra niemal zawsze kończy się remisem. Popęśnienie jakiegokolwiek błędu skutkuje natychmiastową wygraną komputera.

Wraz ze zwiększaniem rozmiaru planszy i ilości symboli potrzebnej do wygrania pojedynku, algorytm Minimax staje się coraz mniej skuteczny. W celu zapewnienia płynności rozgrywki, należy zastosować algorytm analizujący mniejszą ilość ruchów niż przy klasycznej wersji rozgrywki, na skutek czego wygrana z komputerem staje się bardziej realna.

6. Literatura

[1] http://mkarasinski.pl/_cms/files/Adam%20Kujawski%20szachy.pdf (data dostępu: 02.06.2019)

[2] http://wazniak.mimuw.edu.pl/index.php?title=Sztuczna_inteligencja/SI_Modu%C5%82_8_-_Gry_dwuosobowe (data dostępu: 02.06.2019)

[3] <http://informatyka.wroc.pl/node/1409> (data dostępu: 02.06.2019)