



Inteligencja Obliczeniowa

Projekt

Temat: Wykorzystanie algorytmów ewolucyjnych w generowaniu strategii dla gry w warcaby

Autorzy: Krzysztof Wilaszek
Tomasz Wójcik

Opis projektu

Celem opracowanego projektu jest implementacja strategii gry w warcaby przy użyciu algorytmów ewolucyjnych. W ramach prac przygotowano również interfejs graficzny pozwalający na rozegranie partii gry przeciwko zaimplementowanej strategii, tak aby możliwe była weryfikacja poziomu zaawansowania opracowanego rozwiązania.

Opis rozwiązania

Reguły gry

Istnieje wiele różnych odmian warcab, często różniących się regułami w stopniu który wymusza całkowitą zmianę strategii, która z powodzeniem sprawdza się w innej wersji gry. W związku z tym konieczne było przyjęcie jednej wersji jako obowiązującej w tworzonym projekcie. Jej zasady przedstawiają się następująco:

- gra rozgrywana jest na planszy 8 x 8
- każdy gracz rozpoczyna grę z 12 pionkami na planszy
- pierwszy ruch wykonuje osoba grająca białymi pionkami
- w każdej turze obowiązkowe jest maksymalne możliwe bicie (takie aby zbić jak najwięcej figur)
- jeżeli ruch pionka kończy się na ostatniej względem kierunku jego ruchu linii planszy staje się on "damka"
- "damka" może poruszać się w dowolnym kierunku o dowolną liczbę pól (o ile nie wymaga to przeskoczenia dwóch sąsiadujących pionów)
- Zwycięstwo następuje w wyniku zbitia wszystkich pionków przeciwnika lub doprowadzenia do sytuacji w której przeciwnik nie może wykonać ruchu
- Gra kończy się remisem, gdy gracze wykonają 15 ruchów bez zbitia żadnego pionka

Funkcja ewaluacji pozycji

W przygotowanej aplikacji wykorzystano rozwiązanie opisane w publikacji:

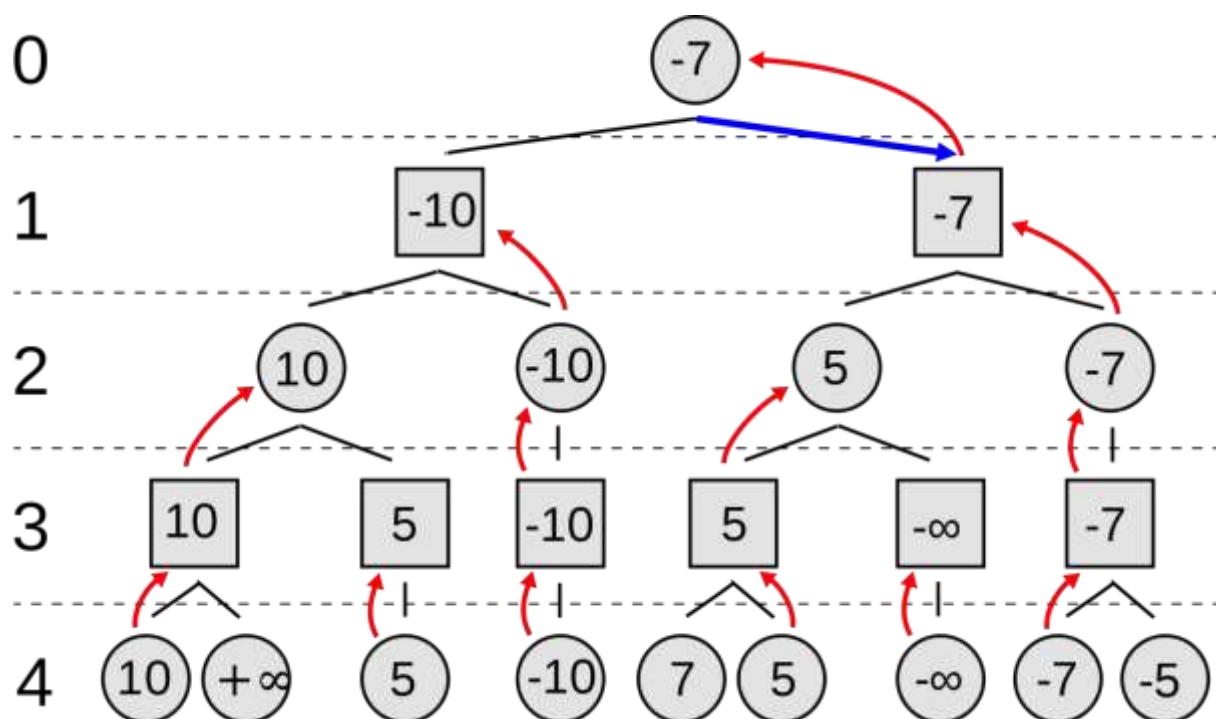
<http://www.mini.pw.edu.pl/~kwaledzik/research/papers/ICANNGA07-Checkers.pdf>.

Na podstawie ustawienia figur na szachownicy obliczane mogą być różne charakterystyki pozycji gracza, np. liczba figur na planszy, różne miary korzyści danego położenia itp. Każda z takich miar posiada swoją wagę odzwierciedlającą istotność danej miary względem pozostałych charakterystyk. Na podstawie liczbowej wartości charakterystyki oraz jej wagi wyliczana jest funkcja ewaluacji danego położenia na szachownicy. Jest to suma iloczynów wagi i wartości miary. Gracz w każdej turze

powinien wybierać taki ruch aby po jego wykonaniu wartość funkcji ewaluacji była dla niego jak najbardziej korzystna. Możliwe jest tutaj również zastosowanie metod przewidywania posunięć przeciwnika, np. wykorzystując algorytm minimax.

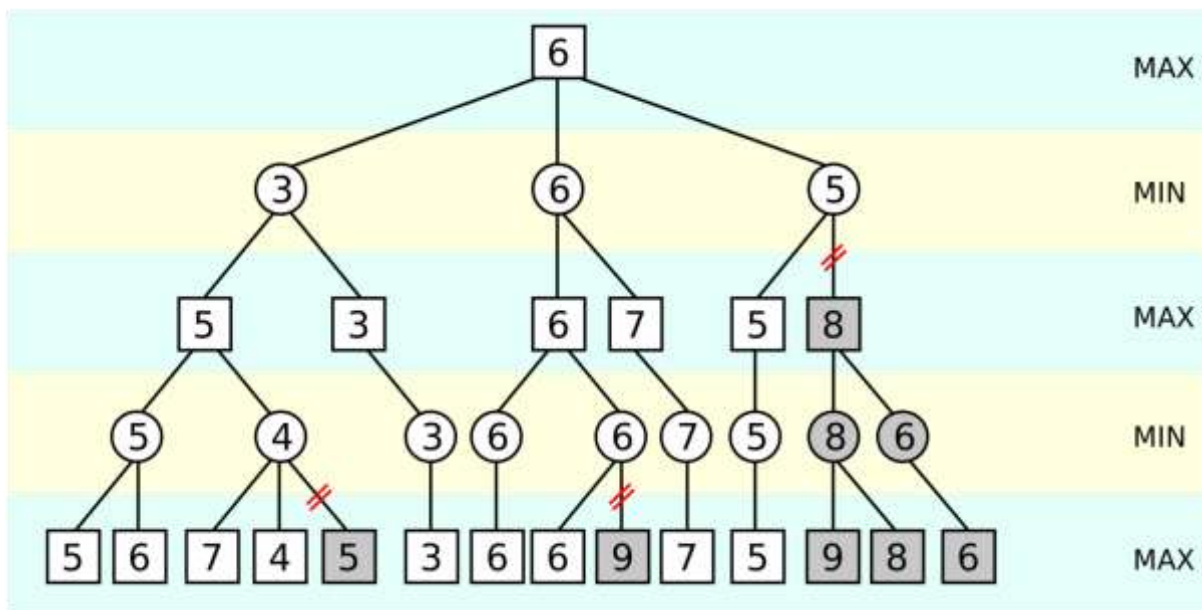
Zastosowanie algorytmu minimax

Algorytm minimax jest prostą metodą służącą do minimalizowania maksymalnych możliwych strat podczas przebiegu gry. Bazuje ona na założeniu, że zarówno gracz kontrolowany przez AI jak i przeciwnik będą w każdej rundzie wybierać najkorzystniejszy dla siebie ruch. Algorytm przeszukuje drzewo możliwych ruchów do wybranej głębokości(w przypadku prostszych gier, np. kółko i krzyżyk jest możliwe przeszukanie drzewa do końca), a następnie oblicza wartość pozycji na planszy wykorzystując podaną funkcję ewaluacji. Obliczone w ten sposób wartości funkcji, opisujące sytuację na planszy po n ruchach są propagowane w górę drzewa. Zgodnie z założeniem o podejmowaniu najlepszych decyzji przez obu graczy, do węzłów na poszczególnych poziomach drzewa są wpisywane najmniejsze(przeciwnik) lub największe(AI) wartości z węzłów potomnych. Schemat działania algorytmu został przedstawiony na poniższym diagramie.



Zastosowanie cięć alfa-beta w algorytmie minimax

Algorytm cięć alfa-beta jest ulepszeniem podstawowego algorytmu minimax, powodującego znaczne przyspieszenie wyszukiwania optymalnego ruchu. Opiera się on na eliminowaniu przeszukiwania tych fragmentów drzewa o których wiemy, że będą gorsze od już znalezionej wartości. Gdy obecna wartość wyszukana w węzłach potomnych jest mniejsza (dla etapu min) lub większa (dla etapu max) od wartości obliczonych dla innego węzła na tym samym poziomie, przeszukiwanie pozostałych potomków rozważanego węzła nie jest konieczne, gdyż wiadomo już, że nie będzie on spełniał kryteriów wyższego poziomu. Mechanizm działania cięć alfa-beta jest przedstawiony na dołączonej ilustracji.



Zastosowanie algorytmu ewolucyjnego

Zadaniem algorytmu ewolucyjnego w przyjętej metodzie jest optymalizacja wag przypisanych każdej z charakterystyk. W tym celu tworzona jest pewna populacja początkowa – zestaw strategii różniących się wartościami współczynników wag dla zastosowanych charakterystyk. Następnie przeprowadzany jest „turniej” – wszystkie strategie zagrają ze sobą. Funkcja przydatności rozwiązania określana jest na podstawie liczby zwycięstw i remisów odniesionych w trakcie „turnieju” (za zwycięstwo lub remis przyznawana jest pewna liczba punktów). Po przeprowadzeniu „turnieju” strategie szeregowane są względem osiągniętej liczby punktów. Następuje tworzenie nowego pokolenia poprzez wykonywanie operacji mutacji i krzyżowania. Prawdopodobieństwo wzięcia udziału w tym procesie uzależnione jest od pozycji w rankingu. Możliwe jest również zachowanie najlepszego lub kilku najlepszych rozwiązań, tak aby nie zostały one utracone w procesach krzyżowania i mutacji. W ten sposób powstaje nowe pokolenie. Rozpoczyna się kolejny etap ewolucji działający według tych samych zasad. Koniec procesu ewolucji następuje po określonej liczbie iteracji.

Najlepsze rozwiązanie(tj. zwycięzca w ostatnim etapie) zostaje zapisane i może być użyte jako przeciwnik dla człowieka lub innych algorytmów gry w warcaby.

Zaimplementowane charakterystyki

W ramach prac przygotowano następujące charakterystyki położenia figur na planszy:

- liczba atakujących pionków – liczba pionków w trzech najbardziej oddalonych od położenia początkowego wierszach
- liczba centralnie położonych pionków – liczba pionków w 8 centralnych polach szachownicy
- liczba centralnie położonych damek
- liczba figur broniących – liczba figur w dwóch najbliższych pozycji startowej wierszach szachownicy
- liczba damek
- liczba pionków
- liczba damek nie sąsiadujących z żadną inną figurą
- liczba pionów nie sąsiadujących z żadną inną figurą
- suma odległości wszystkich pionków gracza od linii promocji
- liczba wolnych pól na linii promocji
- liczba „bezpiecznych” pionów, tzn. znajdujących się na krawędzi szachownicy
- liczba „bezpiecznych” damek
- parametr „triangle” – liczba białych pionków ustawionych w polach E1, F2, G1

Otrzymane wyniki

W ramach prac przeprowadzono testy funkcjonowania aplikacji. Do tego celu przygotowano zestawienie wag parametrów oparte o subiektywną ocenę wagi danej charakterystyki. Założono, że najbardziej istotne parametry to liczba pionków i liczba damek. Istotny powinien być również współczynnik opisujący liczbę bezpiecznych pionków na planszy. Tak przygotowaną strategię skonfrontowano ze strategią wygenerowaną w procesie ewolucyjnym dla różnych liczb przewidywanych w przód ruchów. Wygenerowana strategia grała białymi figurami; strategia oparta o przewidywaną ważność charakterystyk grała czarnymi figurami. Dla każdego testu rozegrano 1000 partii. Otrzymano wyniki :

Liczba przewidywanych ruchów	Liczba zwycięstw białych	Liczba zwycięstw czarnych	Liczba remisów
1	860	90	50
2	867	78	55
3	864	87	49

Tabela 1. Zależność liczby zwycięstw wygenerowanej strategii w pojedynkach z wymyśloną strategią od liczby przewidywanych ruchów.

Wygenerowana strategia wygrywa w znacznej części pojedynków.

W kolejnym teście porównano skuteczność wygenerowanej strategii(białe figury) w pojedynkach z graczem dokonującym losowych ruchów(czarne figury). Otrzymano następujące wyniki:

Liczba przewidywanych ruchów	Liczba zwycięstw białych	Liczba zwycięstw czarnych	Liczba remisów
1	572	362	66
2	554	371	75
3	523	393	84

Tabela 2. Zależność liczby zwycięstw wygenerowanej strategii nad graczem dokonującym losowych ruchów w zależności od liczby przewidywanych ruchów.

Ponieważ algorytm minimax zakłada najlepsze możliwe ruchy przeciwnika w celu minimalizacji maksymalnych strat gracza, nie najlepiej radzi sobie w starciu z przeciwnikiem wykonującym losowe ruchy. W tym przypadku bardziej korzystne byłoby zastosowanie innego algorytmu przeszukiwania, nie zakładającego nic o ruchach przeciwnika.