RAPORT

Systemy wspomagania podejmowania decyzji

Zadanie projektowe:
Algorytm MIN-MAX z funkcją oceniającą
zaimplementowany w grze WARCABY

Przygotował:

Krystian Marecki 222382
Michał Miotk 222331
Wt. TN 15:15
Automatyka i Robotyka
Wydział Mechaniczny

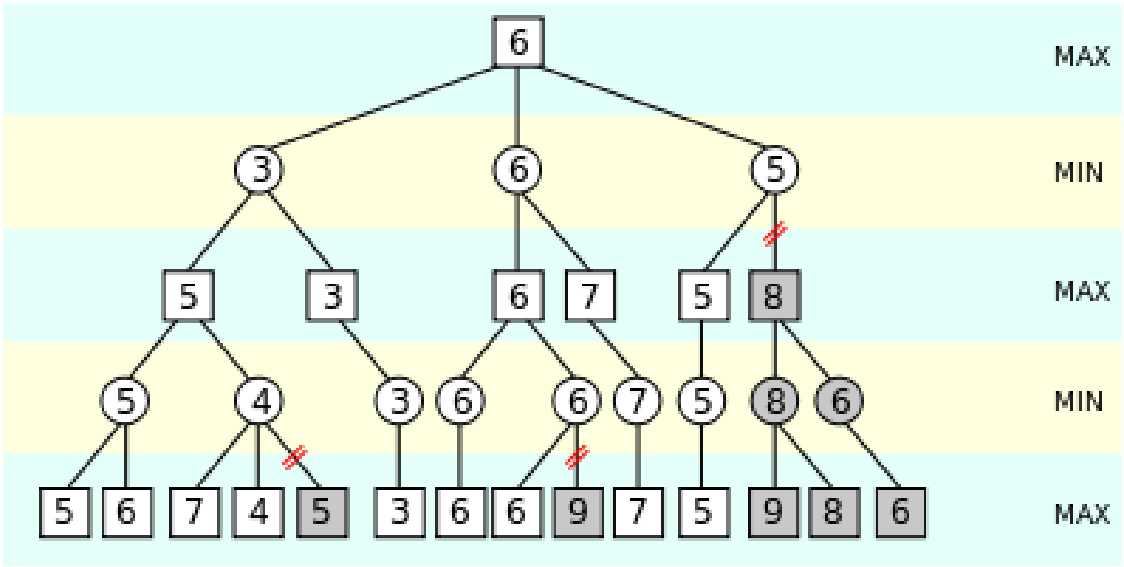
1. Wstęp

Problemem który mieliśmy było opracowanie odpowiedniej strategii gracza w warcabach, a dokładniej programu do samodzielnej gry. Głównym zadaniem było napisanie funkcji oceniającej która współpracowałaby z algorytmem MIN-MAX. Temat ten jest dość powszechny, lecz trudno o sprecyzowane informacje. Tworząc funkcje oceniające polegaliśmy strategiach profesjonalnych graczy. Funkcja ta jest napisana w C++.

2. Algorytm MIN-MAX

Algorytm MIN-MAX polega na przeszukaniu wszystkich możliwych kombinacji ruchów w celu wybrania optymalnego zagrania. Polega on na tym iż po każdej symulacji kilku ruchów, zarówno swoich jak i przeciwnika, ocenia wartość ogólną planszy patrząc na siebie (w naszym przypadku odejmuje wartość planszy gracza 1 od wartości gracza 2), a następnie wybiera raz maksymalną (dla gracza 1) a raz minimalną (dla gracza 2), zakładając że przeciwnik wykona najlepszy ruch dla siebie i w ten sposób wybiera optymalną drogę (dla gracza 1).

Przykładowe przedstawienie graficzne metody prezentuje się następująco:



Algorytm ten mocno obciąża pamięć operacyjną komputera, dlatego warto zastosować obcinanie alfa-beta. Polega na tym iż, jeżeli analizując drzewo gry mamy wykonać ruch po którym stan gry będzie gorszy od najlepszego gwarantowanego, nie przeprowadzamy dalszej analizy tej gałęzi drzewa.

3. Funkcja oceniająca

3.1. Ilość i rodzaj pionków

Podstawowy element oceniający wartość planszy. Jak wiadomo, aby wygrać należy wyeliminować wszystkie pionki przeciwnika. Zliczamy wszystkie pionki oraz damki, poszczególnych graczy, i sumujemy je z odpowiednimi wagami(wagi zostaną opisane niżej). Suma tych jest główną składową wartości funkcji oceniającej

3.2. Znajdowanie bić/wielokrotnych bić

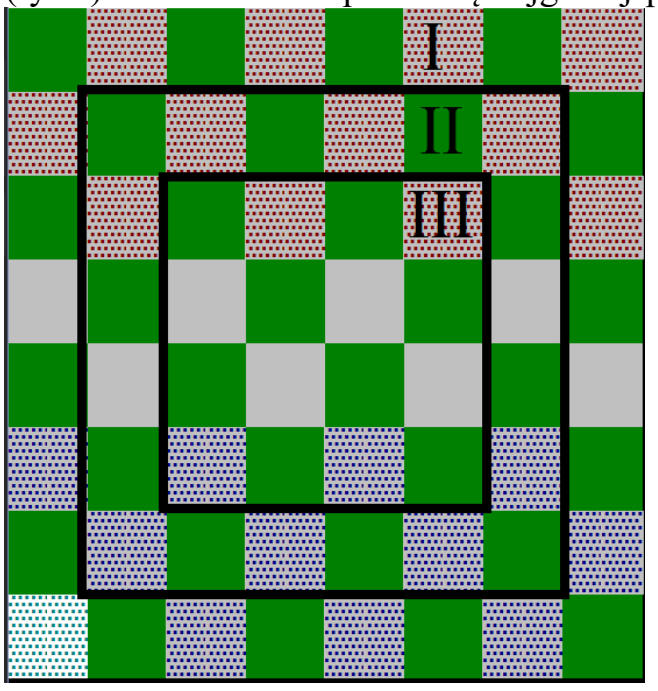
Wiadomo że, aby wygrać należy zbić wszystkie pionki, więc kluczowym elementem jest ich wykrywanie. W tym celu przeszukujemy całą planszę pod kątem znalezienia tak owych. Aby bicie było możliwe pionki muszą stać koło siebie oraz za pionkiem musi być wolne miejsce(w tym miejscu możemy wykonać tą samą funkcje jeszcze raz (zastosować rekurencje) dla konkretnego pionka, aby sprawdzić bicia wielokrotne). Jeśli taki układ występuje do ostatecznej wartości funkcji jest dodawana odpowiednia stała za każde z nich(za wielokrotne odpowiednio więcej).

3.3. Odległość od linii awansu

Wiemy że damki dają dużą przewagę w grze dzięki swojej mobilności, czyli poruszaniu się do tylu i o kilka pól na raz, dlatego ważnym elementem jest ich zdobycie. Awans pionka na damę odbywa się gdy ten dojdzie do przeciwległej krawędzi szachownicy, więc każdy pionek znajdujący się bliżej owej krawędzi jest lepiej punktowany.

3.4. Strefy punktowania

Strefy punktowe promują pionki znajdujące się w środkowej części szachownicy i zapobiega ich poruszaniu się w skrajnych, gdyż zmniejsza to znacznie ich mobilność. Plansza została podzielona na trzy obszary (rys.2). W obszarze I pionki są najgorzej punktowane, a w III najlepiej.

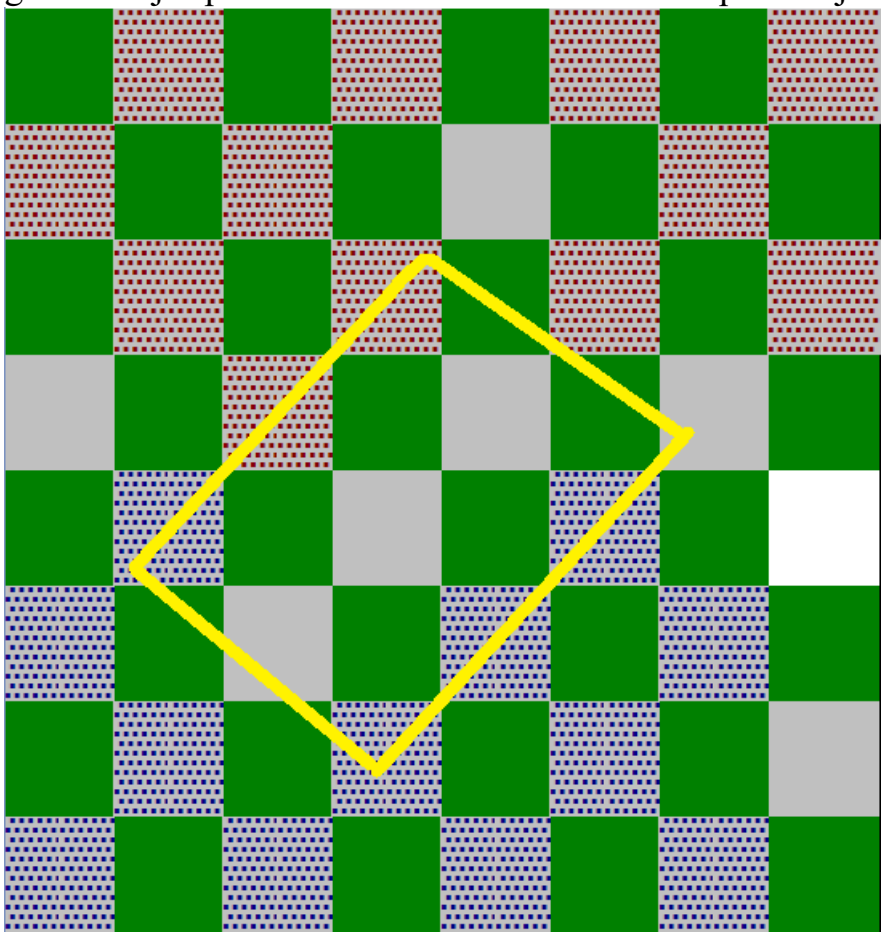


3.5. **Bezpieczeństwo**

Ta zasada polega na tym aby pionki poruszały się tak, aby uniemożliwić bicie przeciwnikowi. Wydawałoby się, że najlepiej zatem poruszać się krawędziami, lecz jak już wcześniej ustaliliśmy nie jest to dobre, gdyż wtedy oddamy przeciwnikowi środkowe pole, co niechybnie doprowadziło by do naszej porażki. Jedyny sposób jaki nam pozostaje to taki aby pionki trzymały się w grupie co eliminuje możliwość ich zbicia. W naszej funkcji ten problem rozwiązaliśmy tak, iż szukamy tak zwanych „sąsiadów", czyli pionków znajdujących się na przyległych pozycjach. Wystarczy że pionek będzie miał dwóch sąsiadów którzy nie znajdują się na przeciwległych pozycjach i jest bezpieczny. Funkcja sprawdza pozycje pionków i przydziela odpowiednią ilość punktów.

3.6. **Zapobieganie korytarzom**

Korytarzem nazywamy ustawienie pionków w dwóch rzędach , między którymi jest luka umożliwiająca przejście pionka bez możliwości zbicia go. Funkcja sprawdza takie ustawienie i obniża punktacje.



3.7. **Blokowanie**

Ważnym aspektem rozgrywki jest blokowanie się pionków, gdyż kiedy nie mamy możliwości dalszego ruchu to przegrywamy oraz każdy zablokowany pionek powoduje ograniczenie naszych opcji wykonania ruchu.

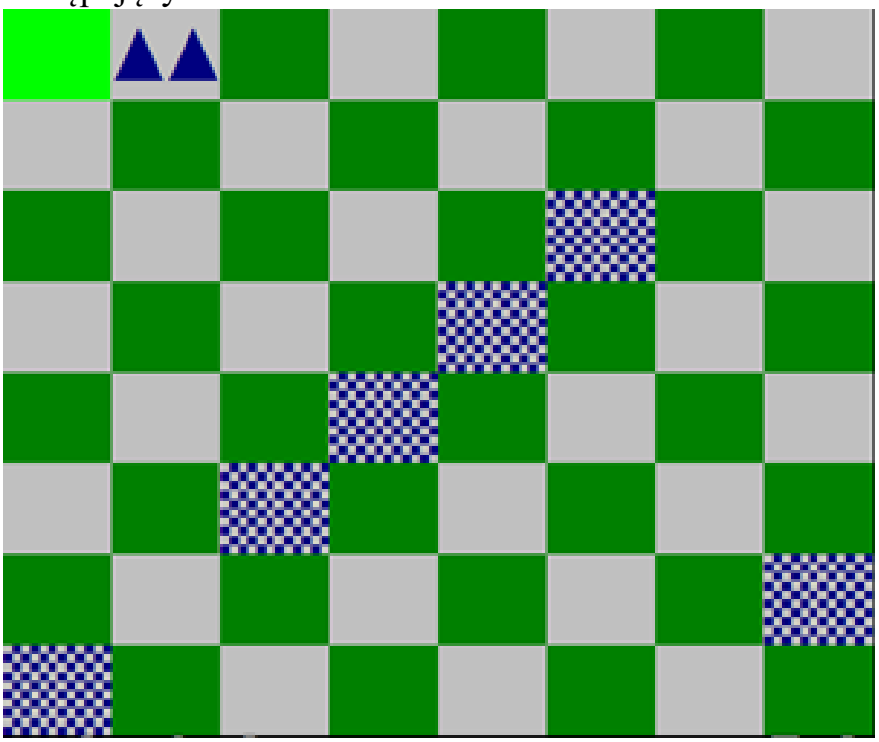
4. Badania

Badania mają na celu dopasowanie odpowiednich wag dla poszczególnych zasad funkcji oceniającej. W tym celu przeprowadziliśmy szereg symulacji. Podstawą naszego założenia było, aby tak dobrać wagi poszczególnych funkcji cząstkowych aby funkcja która ma mniejszą głębokość przeszukiwania wygrała. W naszym przypadku gracz czerwony dla ustawień początkowych miał pokonać niebieskiego:

```
//czerwoni
#define glebokosc_czerwoni 4
#define blizej_konca_czerwoni 2
#define waga_bicie_czerwoni 50
#define chodzenie_grupa_czerwoni 10
#define trzy_srodek_czerwoni 3
#define dwa_srodek_czerwoni 6
#define srodek_czerwoni 9
#define waga_pionek_czerwoni 100
#define chodzenie_dama_czerwoni 10
#define waga_dama_czerwoni 300
//niebiescy
#define glebokosc_niebiescy 5
#define blizej_konca_niebiescy 2
#define waga_bicie_niebiescy 50
#define chodzenie_grupa_niebiescy 10
#define trzy_srodek_niebiescy 3
#define dwa_srodek_niebiescy 6
#define srodek_niebiescy 9
#define waga_pionek_niebiescy 100
#define chodzenie_dama_niebiescy 10
#define waga_dama_niebiescy 300
```

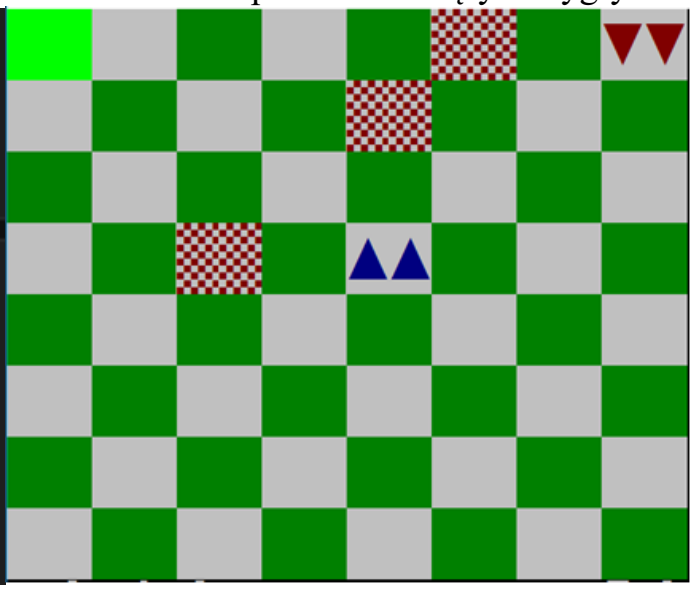
glebokosc_czerwoni–głębokość przeszukiwania dla gracza żółtego
bliżej_konca_czerwoni – punkty za położenie pionka zgodnie z jego kierunkiem ruchu(bliżej końca planszy)
waga_bicie_czerwoni – punkty za bicie w kolejnym ruchu
chodzenie_grupa_czerwoni – punkty za specyficzne położenie swoich pionków wokół rozpatrywanego pionka
trzy_srodek_czerwoni – punkty za położenie 1 pól od lewej lub prawej krawędzi
dwa_srodek_czerwoni – punkty za polozenie 2 pól od lewej lub prawej krawędzi
srodek_czerwoni - – punkty za polozenie 3 pól od lewej lub prawej krawędzi
waga_pionek_czerwoni – punkty za istnienie pionka
chodzenie_dama_czerwoni – punkty za specyficzne położenie dam
waga_dama_czerwoni – punkty za istnienie pionka

Wynik rozegranej partii z powyższymi ustawieniami początkowymi był następujący:



Duże zmniejszenie wagi istnienia pojedynczego pionka w ocenie gracza czerwonego spowodowało że czerwone pionki zaczęły wygrywać:

```
//czerwoni
#define glebokosc_czerwoni 4
#define blizej_konca_czerwoni 2
#define waga_bicie_czerwoni 50
#define chodzenie_grupa_czerwoni 10
#define trzy_srodek_czerwoni 3
#define dwa_srodek_czerwoni 6
#define srodek_czerwoni 9
#define waga_pionek_czerwoni 40
#define chodzenie_dama_czerwoni 10
#define waga_dama_czerwoni 300
//niebiescy
#define glebokosc_niebiescy 5
#define blizej_konca_niebiescy 2
#define waga_bicie_niebiescy 50
#define chodzenie_grupa_niebiescy 10
#define trzy_srodek_niebiescy 3
#define dwa_srodek_niebiescy 6
#define srodek_niebiescy 9
#define waga_pionek_niebiescy 100
#define chodzenie_dama_niebiescy 10
#define waga_dama_niebiescy 300
```



Dalsze zmniejszanie wagi pionka pozwoliło na wygraną czerwonego gracza nawet przy dalszym przewidywaniu(5 poziom) gracza niebieskiego:

Kolejnym krokiem była próba wygrania z większą przewagą – zmieniliśmy jedynie wagi za położenie pionków bliżej środka z 3,6,9 na 10,20,30:

Zauważamy że przewaga czerwonych się zmniejszyła.

Zmieniliśmy wagę za położenie pionków bliżej środka z 10,20,30 na 1,2,3:

Przy tych ustawieniach niebiescy wygrali co wnioskuje że położenie przy środku jest istotnym elementem.

Zwiększenie chodzenie_grupa_czerwoni z 10 na 15 :

pogorszyło wynik – czerwoni wygrali mając o jeden pionek mniej niż poprzednio

Zwiększenie chodzenie_grupa_czerwoni z 15 na 30:

```
//czerwoni
#define glebokosc_czerwoni 4
#define blizej_konca_czerwoni 2
#define waga_bicie_czerwoni 50
#define chodzenie_grupa_czerwoni 30
#define trzy_srodek_czerwoni 3
#define dwa_srodek_czerwoni 6
#define srodek_czerwoni 9
#define waga_pionek_czerwoni 20
#define chodzenie_dama_czerwoni 10
#define waga_dama_czerwoni 300
//niebiescy
#define glebokosc_niebiescy 5
#define blizej_konca_niebiescy 2
#define waga_bicie_niebiescy 50
#define chodzenie_grupa_niebiescy 10
#define trzy_srodek_niebiescy 3
#define dwa_srodek_niebiescy 6
#define srodek_niebiescy 9
#define waga_pionek_niebiescy 100
#define chodzenie_dama_niebiescy 10
#define waga_dama_niebiescy 300
```

nie przyniosło dobrych rezultatów – niebiescy wygrali

Skoro przy wyrównanych szansach w grze jest mało dam zmniejszmy ich wagę z 300 na 100:

```
//czerwoni
#define glebokosc_czerwoni 4
#define blizej_konca_czerwoni 2
#define waga_bicie_czerwoni 50
#define chodzenie_grupa_czerwoni 10
#define trzy_srodek_czerwoni 3
#define dwa_srodek_czerwoni 6
#define srodek_czerwoni 9
#define waga_pionek_czerwoni 20
#define chodzenie_dama_czerwoni 10
#define waga_dama_czerwoni 100
//niebiescy
#define glebokosc_niebiescy 5
#define blizej_konca_niebiescy 2
#define waga_bicie_niebiescy 50
#define chodzenie_grupa_niebiescy 10
#define trzy_srodek_niebiescy 3
#define dwa_srodek_niebiescy 6
#define srodek_niebiescy 9
#define waga_pionek_niebiescy 100
#define chodzenie_dama_niebiescy 10
#define waga_dama_niebiescy 300
```

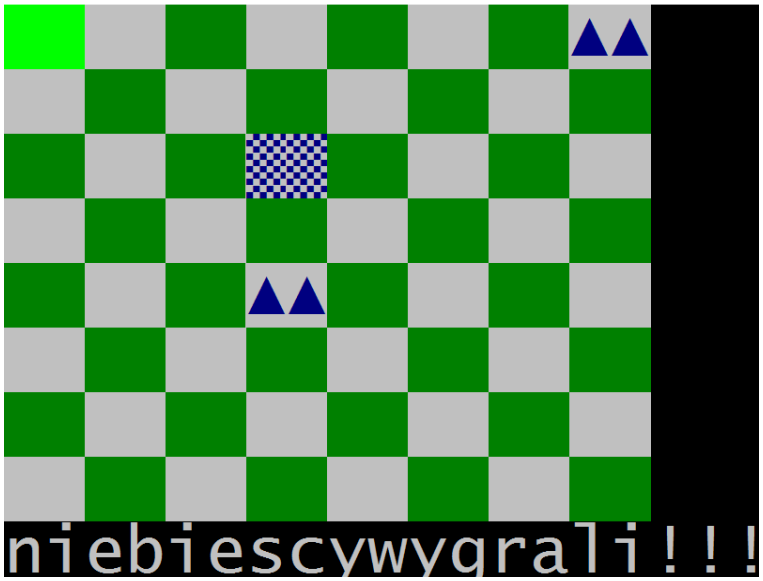
Widać że bilans pionków po wygranej jest podobny jednak pionki czerwone(dla nich zmieniliśmy wagę istnienia dam) są bardziej oddalone od swoich początkowych pozycji niż poprzednio – strategia taka może być dobra do grania człowiek kontra komputer(człowiek często dąży do stworzenia dam).

Dalsze zmniejszanie waga_dama_czerwoni z 100 na 50 nie przyniosło znaczących zmian, zmiana z 50 na 10 tego parametru przyniosła przegraną czerwonych:

```
//czerwoni
#define glebokosc_czerwoni 4
#define blizej_konca_czerwoni 2
#define waga_bicie_czerwoni 50
#define chodzenie_grupa_czerwoni 10
#define trzy_srodek_czerwoni 3
#define dwa_srodek_czerwoni 6
#define srodek_czerwoni 9
#define waga_pionek_czerwoni 20
#define chodzenie_dama_czerwoni 10
#define waga_dama_czerwoni 10
//niebiescy
#define glebokosc_niebiescy 5
#define blizej_konca_niebiescy 2
#define waga_bicie_niebiescy 50
#define chodzenie_grupa_niebiescy 10
#define trzy_srodek_niebiescy 3
#define dwa_srodek_niebiescy 6
#define srodek_niebiescy 9
#define waga_pionek_niebiescy 100
#define chodzenie_dama_niebiescy 10
#define waga_dama_niebiescy 300
```

Próba zmiany parametru bliżej_konca_czerwoni z 2 na 5 spowodowała przegraną czerwonych:

```
//czerwoni
#define glebokosc_czerwoni 4
#define blizej_konca_czerwoni 5
#define waga_bicie_czerwoni 50
#define chodzenie_grupa_czerwoni 10
#define trzy_srodek_czerwoni 3
#define dwa_srodek_czerwoni 6
#define srodek_czerwoni 9
#define waga_pionek_czerwoni 20
#define chodzenie_dama_czerwoni 10
#define waga_dama_czerwoni 25
//niebiescy
#define glebokosc_niebiescy 5
#define blizej_konca_niebiescy 2
#define waga_bicie_niebiescy 50
#define chodzenie_grupa_niebiescy 10
#define trzy_srodek_niebiescy 3
#define dwa_srodek_niebiescy 6
#define srodek_niebiescy 9
#define waga_pionek_niebiescy 100
#define chodzenie_dama_niebiescy 10
#define waga_dama_niebiescy 300
```



Zmniejszenie parametru waga_bicie_czerwoni z 50 na 10 spowodowało klęske czerwonych:

```
//czerwoni
#define glebokosc_czerwoni 4
#define blizej_konca_czerwoni 2
#define waga_bicie_czerwoni 10
#define chodzenie_grupa_czerwoni 10
#define trzy_srodek_czerwoni 3
#define dwa_srodek_czerwoni 6
#define srodek_czerwoni 9
#define waga_pionek_czerwoni 20
#define chodzenie_dama_czerwoni 10
#define waga_dama_czerwoni 25
//niebiescy
#define glebokosc_niebiescy 5
#define blizej_konca_niebiescy 2
#define waga_bicie_niebiescy 50
#define chodzenie_grupa_niebiescy 10
#define trzy_srodek_niebiescy 3
#define dwa_srodek_niebiescy 6
#define srodek_niebiescy 9
#define waga_pionek_niebiescy 100
#define chodzenie_dama_niebiescy 10
#define waga_dama_niebiescy 300
```

Po przeanalizowaniu wszystkich badań i wykonania ich krotności ustaliliśmy że optymalne ustawienia to:

```
//czerwoni
#define glebokosc_czerwoni 4
#define blizej_konca_czerwoni 2
#define waga_bicie_czerwoni 50
#define chodzenie_grupa_czerwoni 10
#define trzy_srodek_czerwoni 3
#define dwa_srodek_czerwoni 6
#define srodek_czerwoni 9
#define waga_pionek_czerwoni 20
#define chodzenie_dama_czerwoni 10
#define waga_dama_czerwoni 25
//niebiescy
#define glebokosc_niebiescy 5
#define blizej_konca_niebiescy 2
#define waga_bicie_niebiescy 50
#define chodzenie_grupa_niebiescy 10
#define trzy_srodek_niebiescy 3
#define dwa_srodek_niebiescy 6
#define srodek_niebiescy 9
#define waga_pionek_niebiescy 100
#define chodzenie_dama_niebiescy 10
#define waga_dama_niebiescy 300
```

5. Wnioski

Napisana przez nas funkcja bez problemu jest zdolna pokonać człowieka przy głębokości przeszukiwania 4/5, natomiast funkcja polegająca tylko na liczeniu pionków potrzebuje głębokości 6. Dzięki ustawieniu odpowiednich wag możemy pokonać algorytm o wyższej głębokości przeszukiwania, co jest bardzo dobrym skutkiem gdyż znacząco skraca obliczenia przy niezmiennej, a nawet lepszej efektywności. Obliczenia przez funkcje nieznacznie wpływają na wydajność obliczeniową, gdyż nie stosujemy rekurencji oraz pętle nie są większe niż dwa zagnieżdżenia . Funkcje uwzględniają większość znanych taktyk gier w warcaby i z pozytywnym skutkiem je wykonuje.