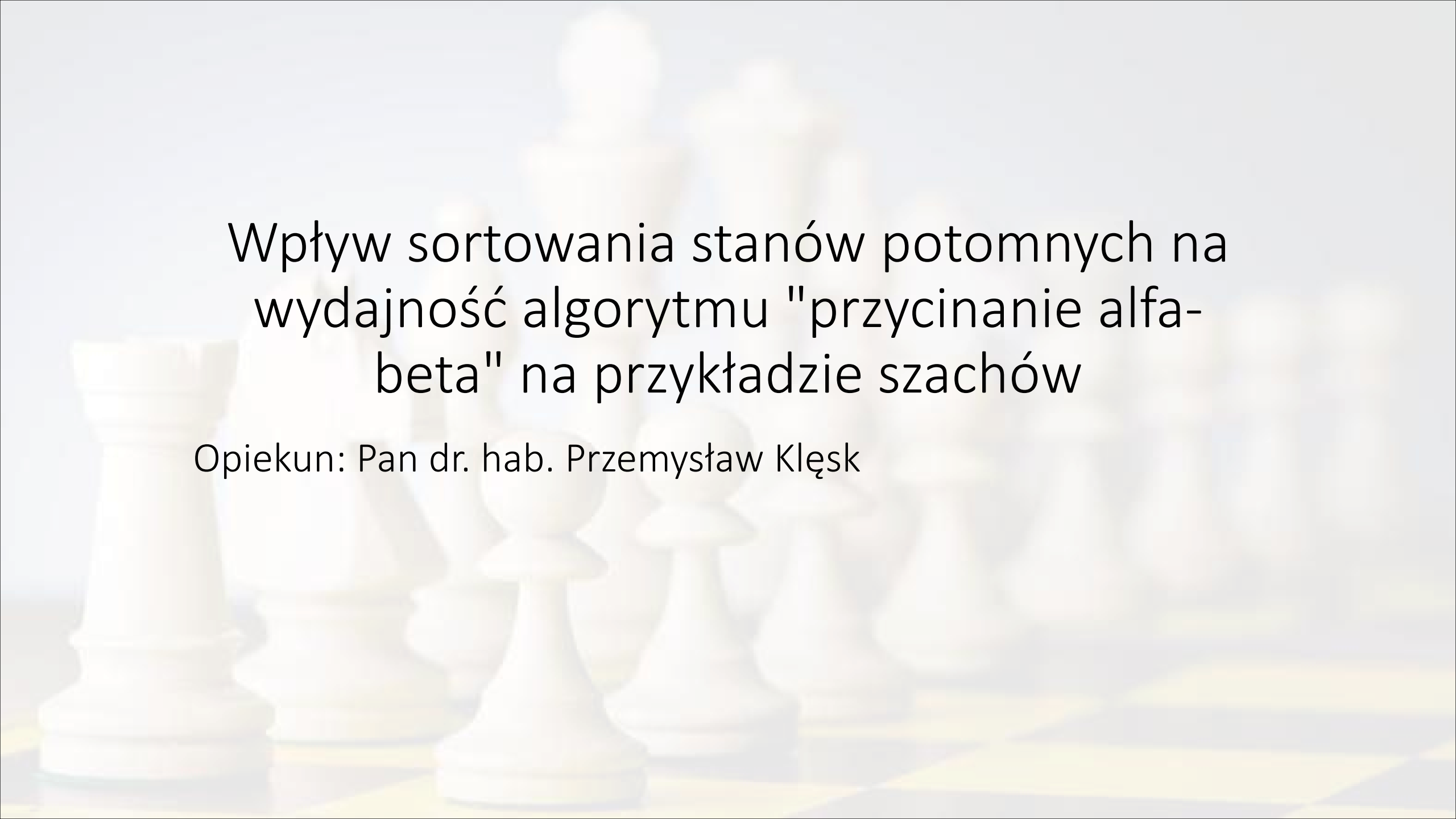


Prezentacja tematu pracy inżynierskiej

Igor Szylak

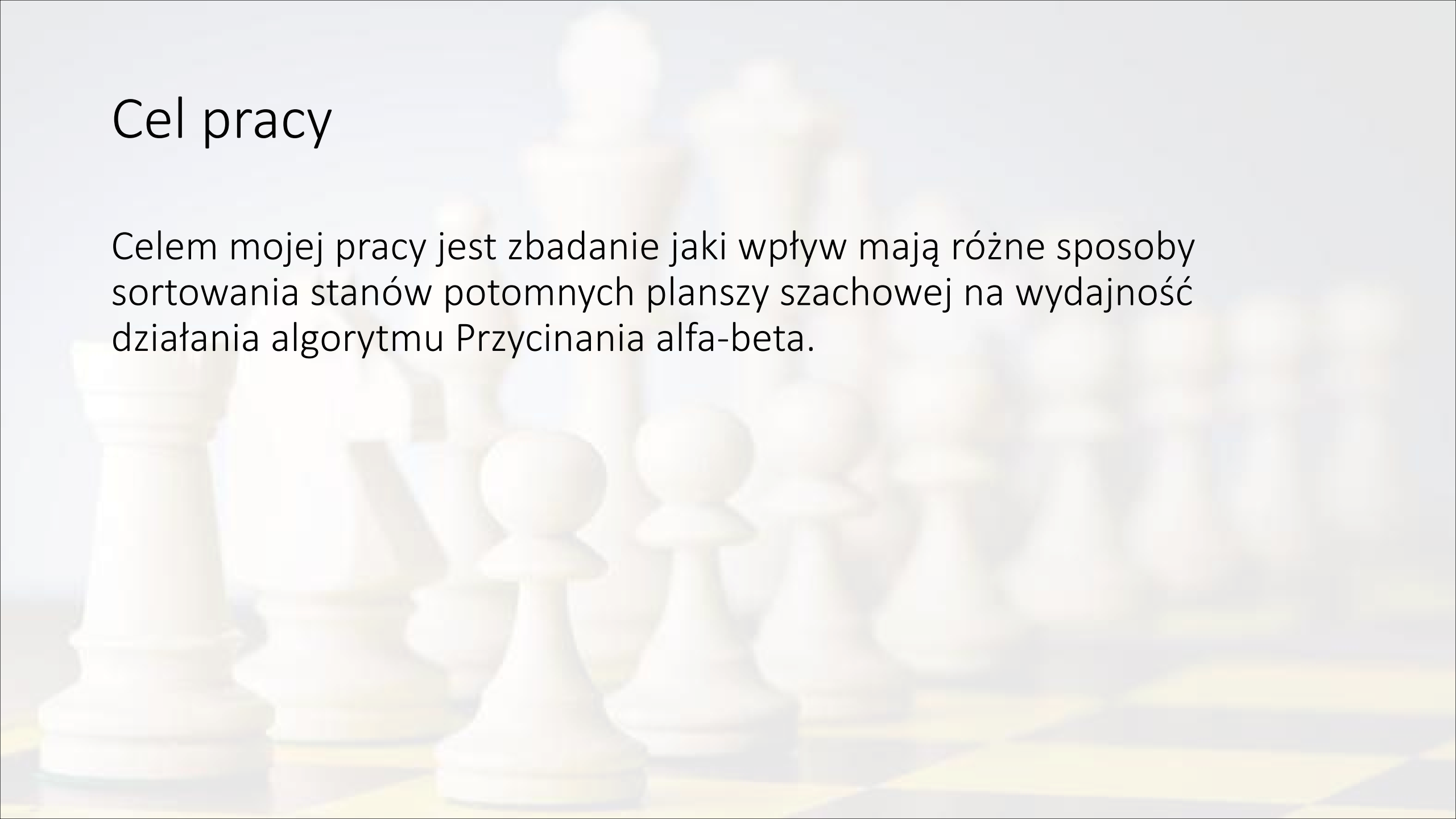


Wpływ sortowania stanów potomnych na wydajność algorytmu "przycinanie alfa-beta" na przykładzie szachów

Opiekun: Pan dr. hab. Przemysław Klęsk

Cel pracy

Celem mojej pracy jest zbadanie jaki wpływ mają różne sposoby sortowania stanów potomnych planszy szachowej na wydajność działania algorytmu Przycinania alfa-beta.



Zakres pracy obejmuje

- Stworzenie silnika szachowego,
- implementacje algorytmów Minimax oraz Przycinania alpha-beta,
- określenie sposobów sortowania stanów potomnych do ewaluacji przez wyżej wymienione algorytmy,
- ocenę tychże sposobów pod względem efektywności i szybkości stworzonej sztucznej inteligencji

Silnik szachowy

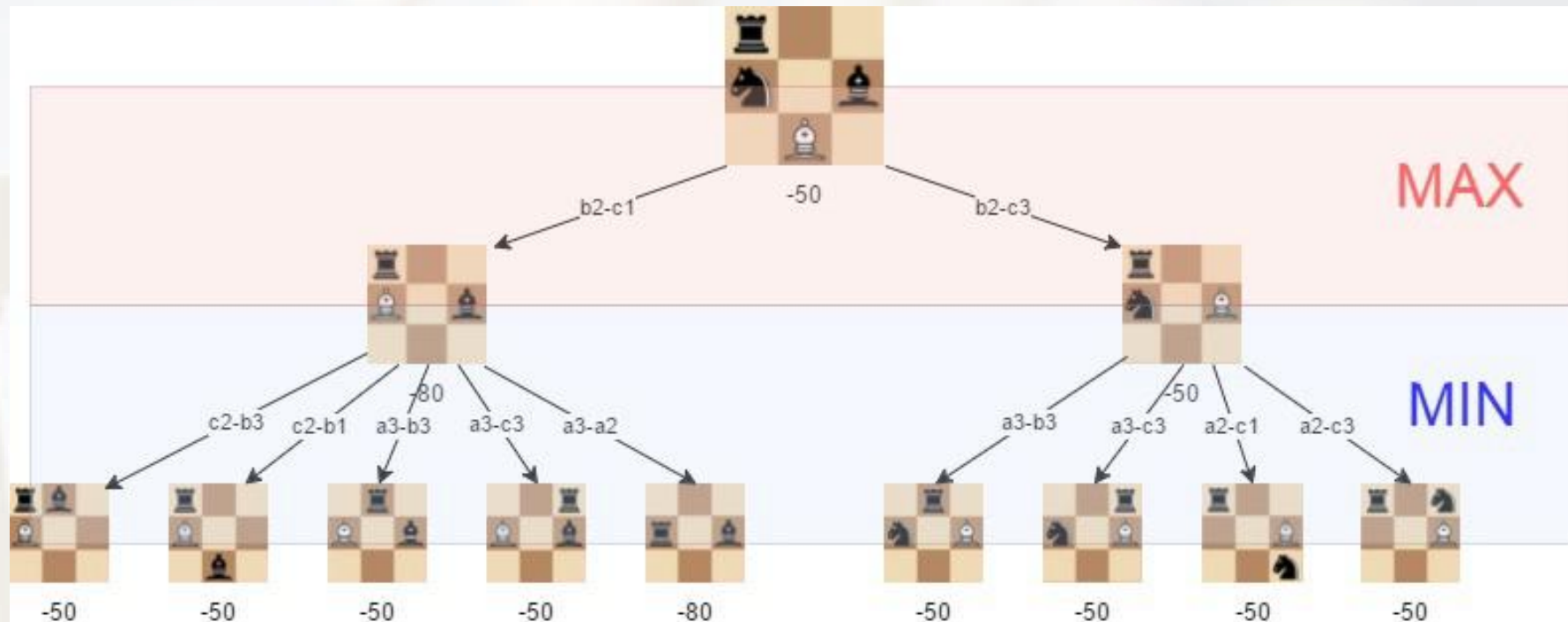
Plansza szachowa zawierająca 64 pola będzie reprezentowana za pomocą dwuwymiarowej macierzy pozycji figur. Silnik wymaga implementacji wszystkich zasad szachowych: rodzajów figur takich jak hetman czy skoczek, ich początkowej pozycji na planszy, ruchów jakie może wykonać dana figura, koncepcji *szacha* czy *mata*.

Ponadto trzeba będzie zadbać o zasady “specjalne” takie jak *roszady* królem, możliwy podwójny pierwszy ruch pionem czy chociażby bicie w przelocie. Ważną sprawą będzie również ocena pozycji np. stworzenie dla każdego rodzaju bierki macierzy przechowującej punkty “siły” bierki w zależności od pola na jakim stoi.

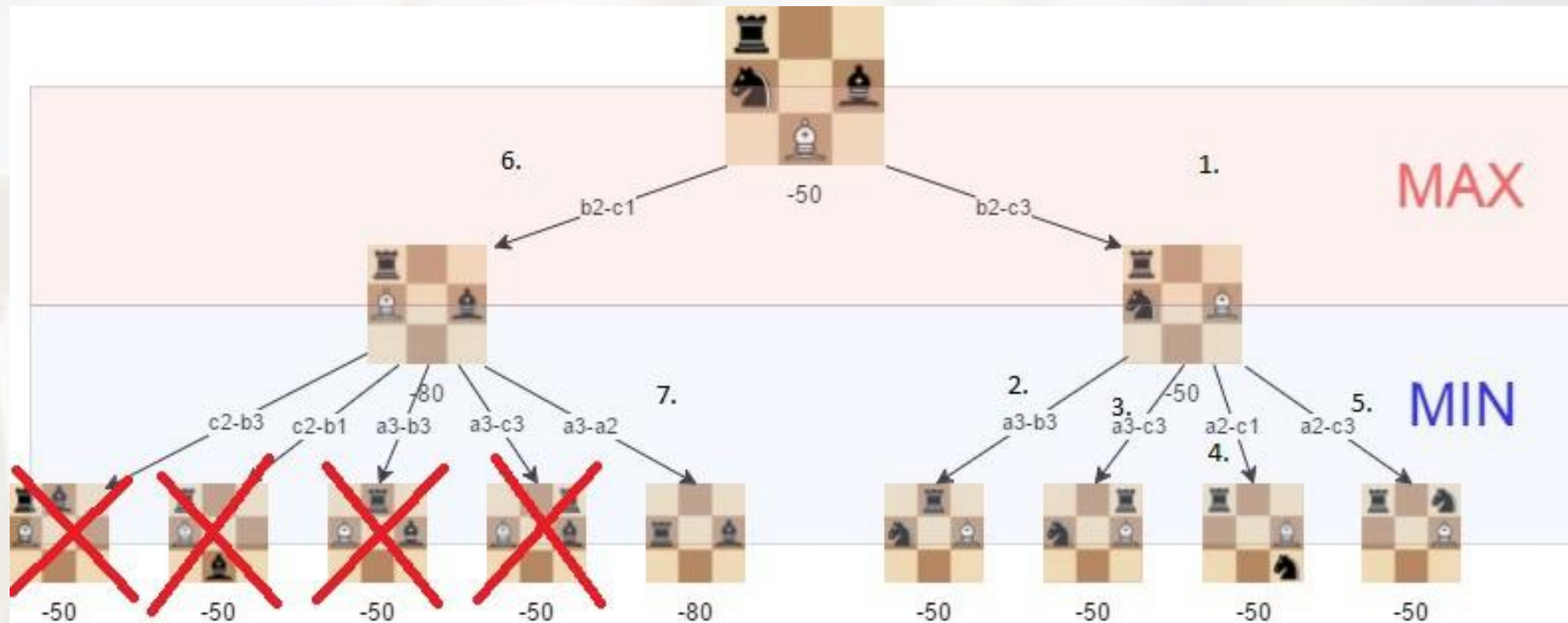
Algorytmy Minimax oraz Przycinanie alpha-beta

- Algorytm Minimax polega na minimalizowaniu maksymalnych strat w grach o sumie zerowej (uogólniając są to wszystkie gry w których jakiś zysk jednego gracza wiąże się z jakąś stratą drugiego gracza) jaką niewątpliwie są szachy.
- Algorytm Przycinania alpha-beta usprawnia pracę przeszukiwania drzewa gry przez Minimax odcinając poddrzewa o których wiadomo, że nie poprawią wartości węzła leżącego wyżej zatem odrzucenie tych poddrzew nie wpłynie na ostateczny wynik oceny.

Jak działa Minimax



Jak działa Przycinanie alpha-beta



Pseudokod algorytmu Przycinania alpha-beta

```
funkcja minimax(węzeł, głębokość)
  zwróć alfabet(węzeł, głębokość,  $-\infty$ ,  $+\infty$ )
```

```
funkcja alfabet(węzeł, głębokość,  $\alpha$ ,  $\beta$ )
  jeżeli węzeł jest końcowy lub głębokość = 0
    zwróć wartość heurystyczną węzła
  jeżeli przeciwnik ma zagrać w węźle
    dla każdego potomka węzła
       $\beta := \min(\beta, \text{alfabet}(\text{potomek}, \text{głębokość}-1, \alpha, \beta))$ 
      jeżeli  $\alpha \geq \beta$ 
        przerwij przeszukiwanie {odcinamy gałąź Alfa}
    zwróć  $\beta$ 
  w przeciwnym przypadku {my mamy zagrać w węźle}
    dla każdego potomka węzła
       $\alpha := \max(\alpha, \text{alfabet}(\text{potomek}, \text{głębokość}-1, \alpha, \beta))$ 
      jeżeli  $\alpha \geq \beta$ 
        przerwij przeszukiwanie {odcinamy gałąź Beta}
    zwróć  $\alpha$ 
```

Sposoby sortowania

Sposoby sortowania pozostają kwestią do rozważenia, przykładowo jeden ze sposobów może opierać się na zapewnieniu pierwszeństwa ruchom zbijającym, a inny priorytetować ruchy szachujące króla przeciwnika.

Kwestia plagiatu

Mając na uwadze mnogość już istniejących prac i artykułów podejmujących temat szachów oraz algorytmów z nimi związanych, nasuwa się wniosek, że musimy być bardzo ostrożni w korzystaniu z różnych źródeł a także uważać by nie zapomnieć umieszczać źródeł i autora przy każdym cytowanym fragmencie.

A collection of chess pieces on a wooden board. On the left is a large white king piece. In the center is a black king piece. In the foreground, slightly to the right of center, is a white pawn. To the right of the pawn is a black knight piece. Further to the right is another white king piece. The text 'Dziękuję za uwagę' is overlaid in the center of the image.

Dziękuję za uwagę