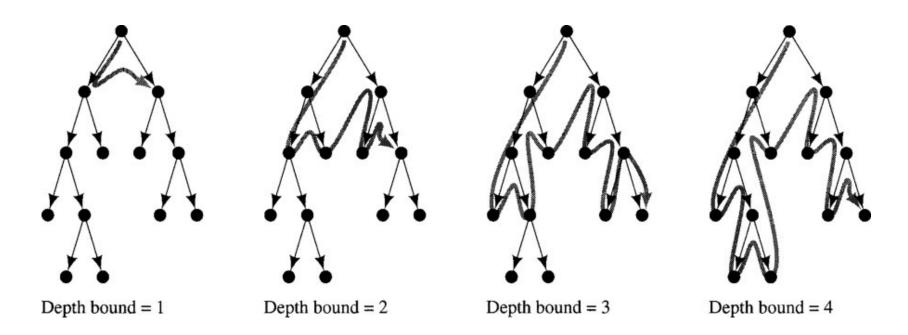
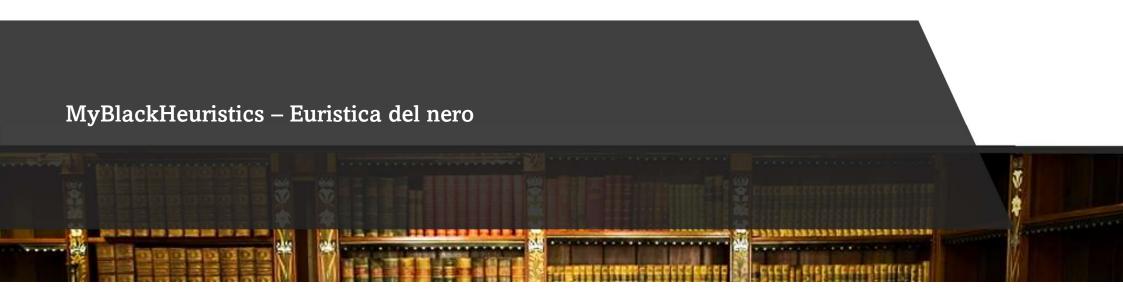


Iterative deepening depth-first search con Alpha-Beta Pruning Implementata usando la libreria AIMA





- ❖ BLACK_ALIVE: Numero di pedine nere sulla scacchiera.
- **❖** WHITE_EATEN: Numero di pedine bianche mangiate dal nostro giocatore.
- ❖ BLACK_SURROUND_KING: Numero di pedine nere che circondano il re.
- **❖** RHOMBUS_POSITIONS: Matrice delle posizioni favorite dei neri per bloccare le vie di fuga.
- Controllo della prossima mossa del vincitore: esamina gli scacchi matti nella prossima mossa



- **❖** WHITE_ALIVE: Numero di pedine bianche sulla scacchiera.
- **❖** BLACK_EATEN: Numero di pedine nere mangiate dal nostro giocatore.
- ***** KING_PROTECTION: Posizioni che garantiscono la sicurezza del re.
- **❖** NUM_ESCAPES_KING: Numero di vie di fuga vittoriose del re.
- **❖** BEST_POSITIONS: Le posizioni migliori per le pedine bianche a inizio partita.
- Controllo delle prossime mosse del vincitore: esamina gli scacchi matti nelle prossime mosse.

MyWhiteHeuristics – Euristica del bianco



Monte Carlo Tree Search

L'approccio iniziale che abbiamo adottato è stato quello di usare l'algoritmo di Montecarlo che si basa sull'idea di simulare delle partite randomiche nel tentativo di trovare il maggior numero di playouts vincenti.

Questa strategia per il poco tempo a disposizione per la simulazione non riesce a restituire un risultato soddisfacente.

