

Chess Promotion Game – Artificial Intelligence

Andrea Sghedoni¹

¹ Università degli studi di Bologna, LM Informatica, Maggio 2015

Abstract. Progetto del corso di Intelligenza Artificiale, Laurea Magistrale in Informatica, I Anno, Luglio 2015, Università degli studi di Bologna (*Alma Mater Studiorum*).

Keywords: Chess, Promotion, AI, MinMax, AlphaBetaPruning.

1 Introduzione

Il progetto ha come scopo l'implementazione di un gioco basato sugli scacchi, in modo da studiare alcuni aspetti peculiari degli ambiente multiagente competitivi. Si ricorda che in questi ambienti si hanno due o più agenti che entrando in competizione tra di loro, cercano tramite una propria funzione di valutazione di massimizzare il proprio benessere, a discapito degli agenti avversari. Il gioco, le cui semplici regole sono spiegate nel capitolo successivo, è a turni e per esplorare lo spazio degli stati del gioco viene utilizzato l'algoritmo MinMax (versione con la Potatura AlphaBeta). Questo dovrebbe ritornare la mossa migliore che il giocatore può effettuare, in base alla mossa dell'avversario e della scacchiera corrente. Si utilizza il condizionale perché visitare tutto lo spazio degli stati (anche se il gioco può apparire semplice e limitato) non è ragionevole. Basti pensare che per arrivare alla conclusione di una partita di scacchi sarebbe necessario un albero con circa 10^{40} nodi (per questo si utilizza una versione a profondità limitata e con potatura Alpha Beta).

2 Regole di Chess Promotion Game

Le regole di questo gioco, come già anticipato, sono molto semplici. L'obiettivo è quello di mandare a promozione un pedone prima dell'avversario, in altre parole il primo che va a promozione vince. La

scacchiera di gioco è semplificata rispetto quella consueta, ovvero, per giocatore, sono presenti solo gli 8 pedoni ed i rispettivi Re. Le regole per questi pezzi sono le solite degli scacchi (mossa En Passant compresa).

3 Algoritmo AlphaBeta Pruning

Il grande problema dell'algoritmo MinMax è quello dell'esplosione del numero di stati, crescente in modo esponenziale con la profondità dell'albero. L'intenzione della potatura Alpha Beta è quello, quantomeno, di limitare questa esplosione, cercando di dimezzare l'esponente che governa l'esponenziale della complessità temporale ($O(b^m)$ con $b \rightarrow$ fattore di branching ed $m \rightarrow$ profondità massima dell'albero).

L'idea di base è quella di accorgersi quando un sottoalbero dello spazio degli stati non verrà mai preso in considerazione come mossa migliore, nè per il player Max, nè per il player Min.

Risulta inutile continuare la ricerca in quel sottoalbero e quindi la valutazione continua da un'altra porzione di albero.

I parametri Alpha e Beta servono proprio come sentinelle per le mosse migliori correnti per il player Max e player Min, rispettivamente.

Grazie al continuo aggiornamento di questi parametri, se una mossa riguardante un sottoalbero non può in nessun modo migliorare quelle corrente, il sottoalbero corrispondente viene tagliato.

Di seguito viene illustrato lo pseudocodice dell'algoritmo che è stato utilizzato nell'implementazione:

```
function alphabetamax(stato, alpha, beta, depth):  
    if TestTerminazione(stato) or (depth == 0):  
        return h(stato)  
    v = -∞  
    for s in SUCESSORE(stato):  
        v = MAX(v, alphabetamin(s, alpha, beta, depth - 1))  
        if v ≥ beta : return v  
        alpha = MAX(v, alpha)  
    return v
```

```

function alphabetamin(stato, alpha, beta, depth):
    if TestTerminazione(stato) or (depth == 0):
        return h(stato)
    v =  $+\infty$ 
    for s in SUCESSORE(stato):
        v = MIN(v, alphabetamax(s, alpha, beta, depth - 1))
        if v  $\leq$  alpha : return v
        beta = MAX(v, beta)
    return v

```

Con una profondità settata a 5, la chiamata iniziale diventa:

```

alphabetamax(stato,  $-\infty$ ,  $+\infty$ , 5)

```

4 Funzione di valutazione

Quando la profondità è giunta al limite, oppure il test terminazione decreta la fine del gioco, si torna una valutazione della tabella corrente. Queste valutazioni, risalendo ricorsivamente l'albero dalle foglie alla radice chiamante, decretano la mossa migliore da effettuare per il player Max (nel caso la chiamata iniziale sia la medesima del Cap. 3).

La funzione di valutazione implementata cerca di fare un'analisi più completa possibile della situazione di gioco corrente.

I parametri considerati sono i seguenti:

- **+10** punti per ogni pedone ancora in gioco
- **-10** punti per ogni pedone avversario in gioco
- punti incrementali per ogni pedone in base alla posizione, da **+0** (posiz. iniziale) a **+100** (posiz. vittoria)
- punti incrementali per ogni pedone avversario in base alla posizione, da **-0** (posiz. iniziale) a **-100** (posiz. vittoria)
- score dimezzato per ogni pedone circondato ed in pericolo di cattura
- score raddoppiato per possibile cattura
- **$+\infty$** per stato promozione
- **$-\infty$** per stato promozione avversario

4 Implementazione

Il gioco è stato implementato con il linguaggio *Python*, il quale si presta bene per problemi di AI.

L'interfaccia grafica è stata implementata con le librerie PyQt4, le quali sono facilmente scaricabili tramite il comando (OS Ubuntu, Linux kernel) :

```
$ sudo apt-get install python-qt4-dev
```

Per l'avvio del gioco basta posizionarsi all'interno della cartella e dare il comando:

```
$ python logic.py
```

Essendo Python un linguaggio interpretato, non ha bisogno di precompilazioni.

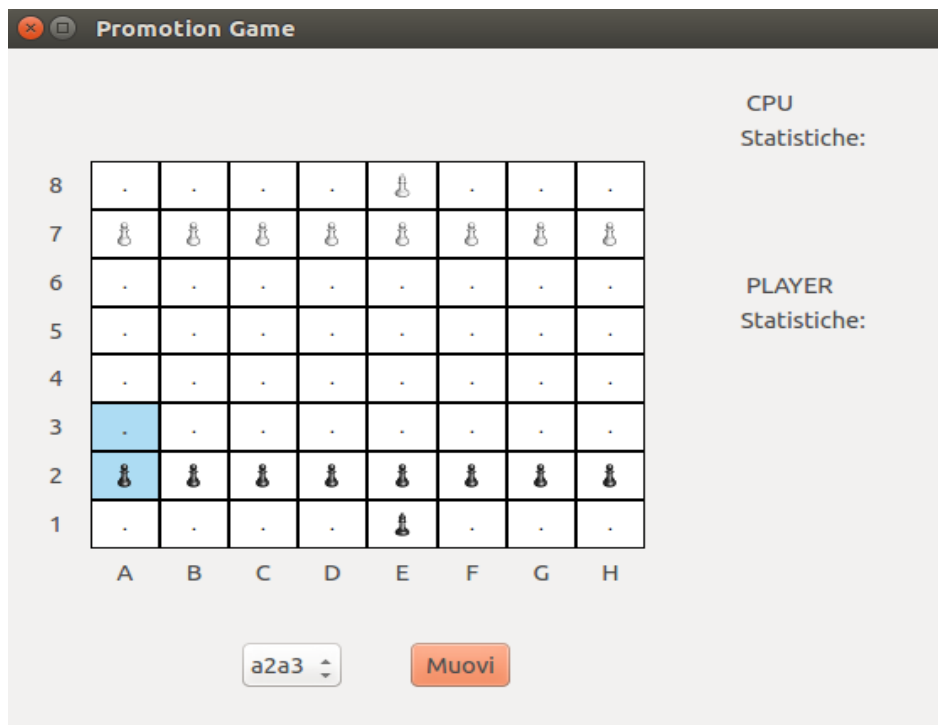


Figura 1 | Interfaccia grafica del gioco Chess Promotion Game

La giocabilità è veramente molto semplice, in quanto basta scegliere una mossa dal menù a tendina e premere il tasto "Muovi" per effettuare la mossa. Da notare che, allo scorrere del mouse sulle mosse possibili nel menù a tendina, queste vengono illuminate sulla scacchiera (caselle azzurre di *figura 1*) per avere un effetto visivo di dove potrebbe muoversi il pezzo.

9 Bibliografia

1. S.Russell, P.Norving, "*Artificial Intelligence, A Modern Approach*", Third Edition. Nov. 2009
2. <https://chessprogramming.wikispaces.com/> , portale di regole, linee guida, programmazione scacchi. Dagli algoritmi alle tabelle di valutazione.
3. <https://wiki.python.org/moin/PyQt4> , sito per la configurazione e l'utilizzo delle librerie PyQt4.