

ALGORITMO MINIMAX

Definizione

L'algoritmo minimax è un algoritmo ricorsivo per la ricerca della mossa migliore in un gioco a somma zero che si svolge tra due giocatori.

Utilizzo

L'algoritmo minimax consente di individuare le scelte migliori dei due giocatori nel corso del gioco, analizzando a ritroso l'albero di gioco a partire dai nodi terminali, ossia dalle possibili situazioni in cui può terminare il gioco (fine gioco), e risalendo progressivamente fino alla posizione corrente dei giocatori.

Obiettivi

L'albero di ricerca si sviluppa su 2 giocatori:

- MAX (noi)
- MIN (l'avversario).

L'obiettivo è raggiungere uno stato terminale di quest'albero con la massimizzazione dell'utilità.

Funzionamento

L'algoritmo è composto da 4 passi:

1. Genera l'albero completo del gioco.
2. Applica la funzione di utilità a tutti gli stati terminali (foglie)
3. Partendo dalle foglie assegna ricorsivamente agli stati intermedi il massimo della funzione di utilità degli stati figli se i figli corrispondono a una mossa di Max.
4. Il minimo della funzione di utilità degli stati figli se i figli corrispondono a una mossa di Min.

L'algoritmo termina con l'assegnazione a Max della prima mossa, che corrisponde al massimo della funzione di utilità al primo livello.

Si applica la funzione di utilità ai nodi terminali (foglie).

Funzione di utilità

La funzione di utilità è una funzione che associa ad ogni combinazione di fine partita un indice di soddisfazione (utilità ordinale) o un valore numerico (utilità cardinale).

I panieri con indice numerico superiore forniscono una utilità maggiore all'algoritmo e viceversa.

Nel caso del tris, le combinazioni che portano alla vittoria da parte dell'algoritmo sono contrassegnate rispettivamente da 1 (vittoria), 0 (pareggio) e -1 (sconfitta).

Problemi di MiniMax

- Il numero di turni deve essere finito
- La complessità computazionale aumenta esponenzialmente in base al numero di turni
- Non è adatto nei giochi con più di 2 giocatori
- Impossibilità di generare (dal punto di vista della complessità computazionale e quindi del tempo di esecuzione), anche per giochi molto semplici, l'albero completo delle mosse.

Ottimizzazioni

Da tutto quello esposto fino ad ora, emerge quindi che l'algoritmo minimax lavora in alberi secondo certe proprietà matematiche.

Perciò considerando anche mosse e nodi che non si verificheranno mai.

Si deve cercare di ridurre lo spazio di ricerca.

La tecnica più conosciuta è quella dell'**alpha-beta pruning** coadiuvato da **funzioni euristiche di valutazione**.

Funzione euristica di valutazione

L'algoritmo Minimax si basa sull'ipotesi che sia possibile generare tutti gli stati terminali.

Questo è di solito impraticabile, anche per giochi non molto complicati.

È necessario un criterio che faccia generare una mossa all'algoritmo Minimax senza dover esplorare tutto l'albero di ricerca.

Questo si può ottenere introducendo una funzione euristica di valutazione, che valuterà la bontà anche di stati non terminali, e un test di terminazione per decidere a che livello di mosse terminare la ricerca.

Lo scopo principale della funzione euristica consiste nel guidare il processo di ricerca lungo il cammino più promettente.

La funzione euristica consente di ridurre la complessità dell'algoritmo sia dal punto di vista del tempo di ricerca (complessità temporale) che dello spazio di memoria (complessità spaziale).

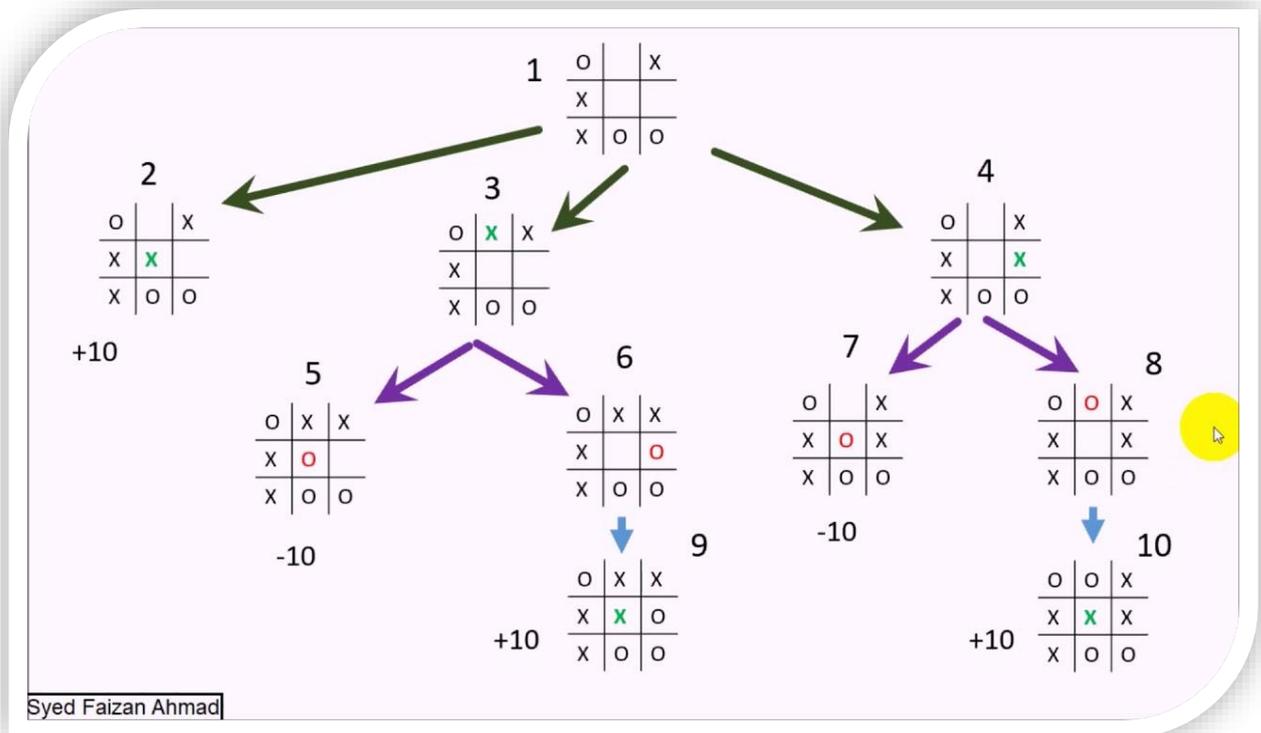
Una funzione di valutazione in genere si basa sull'importanza delle mosse, che non è costante, ma varia in base alla posizione e alla fase della partita.

La funzione di valutazione dà una stima della bontà della mossa.

IL NOSTRO PROGETTO

Andremo a realizzare una semplice applicazione, più in particolare il gioco del tris che implementerà l'algoritmo MiniMax.

Evidenzieremo inoltre, la differenza in termini di prestazioni (spaziali e temporali) tra MiniMax puro e MiniMax con Alpha – Beta Pruning.



- Pseudocodice MiniMax

```

function minimax(nodo, profondità)
  SE nodo è un nodo terminale OPPURE profondità = 0
    return il valore euristico del nodo
  SE l'avversario deve giocare
     $\alpha := +\infty$ 
    PER OGNI figlio di nodo
       $\alpha := \min(\alpha, \text{minimax}(\text{figlio}, \text{profondità}-1))$ 
  ALTRIMENTI dobbiamo giocare noi
     $\alpha := -\infty$ 
    PER OGNI figlio di nodo
       $\alpha := \max(\alpha, \text{minimax}(\text{figlio}, \text{profondità}-1))$ 
  return  $\alpha$ 
  
```

- Pseudocodice Alpha - Beta Pruning

```

01 FUNZIONE alfa_beta(nodo, profondità,  $\alpha$ ,  $\beta$ , massimizza)
02   SE profondità = 0 O nodo è terminale
03     RESTITUISCI valore euristico del nodo
04   SE massimizza
05      $v := -\infty$ 
06     PER OGNI figlio del nodo
07        $v := \max(v, \text{alfa\_beta}(\text{figlio}, \text{profondità} - 1, \alpha, \beta, \text{FALSO}))$ 
08        $\alpha := \max(\alpha, v)$ 
09       SE  $\beta \leq \alpha$ 
10         INTERROMPI IL CICLO (* taglio secondo  $\beta$  *)
11     RESTITUISCI  $v$ 
12   ALTRIMENTI
13      $v := +\infty$ 
14     PER OGNI figlio del nodo
15        $v := \min(v, \text{alfa\_beta}(\text{figlio}, \text{profondità} - 1, \alpha, \beta, \text{VERO}))$ 
16        $\beta := \min(\beta, v)$ 
17       SE  $\beta \leq \alpha$ 
18         INTERROMPI IL CICLO (* taglio secondo  $\alpha$  *)
19     RESTITUISCI  $v$ 

(* richiamo iniziale *)
alfa_beta(origine, profondità,  $-\infty$ ,  $+\infty$ , VERO)
  
```

BIBLIOGRAFIA - SITOGRAFIA

- **Definizione & Utilizzo** - https://www.okpedia.it/algoritmo_minimax
- **Obiettivi** - [Intelligenza Artificiale - GamePlaying.pdf](#)
- **Funzionamento** - [risoluzione d 2003.ppt](#) [risoluzione d 2003.pdf](#)
- **Funzione di utilità** - [https://www.okpedia.it/funzione di utilita](https://www.okpedia.it/funzione_di_utilita)
- **Funzione euristica** - [https://www.okpedia.it/funzione euristica](https://www.okpedia.it/funzione_euristica)
- **Ottimizzazioni** - <https://docplayer.it/24705033-Max-max-min-albero-di-gioco.html>
- **Alpha – Beta Pruning** - https://it.wikipedia.org/wiki/Potatura_alfa-beta
- **Pseudocodice MiniMax** - <https://it.wikipedia.org/wiki/Minimax>
- **Pseudocodice A – B Pruning** - https://it.wikipedia.org/wiki/Potatura_alfa-beta