

Areaperta tra gioco e scienza:

La sfida tra supercomputer e campione mondiale

Maurizio Parton, Università di Chieti-Pescara
Fabrizio Falchi, ISTI CNR, Pisa
Francesco Potortì, ISTI CNR, Pisa

Il Go è uno dei giochi astratti più antichi del mondo, e da decenni l'intelligenza artificiale si cimenta nel tentativo di giocare alla pari con l'essere umano.

Nel gennaio 2016 il software AlphaGo di Google annuncia la sua vittoria contro il pluricampione europeo. Resta ancora un'ultima sfida: è previsto per il mese di

Cos'è un gioco?



Un *gioco* è una situazione in cui i *giocatori* possono compiere delle scelte, per ricevere una *ricompensa*.


Elementi caratterizzanti:

- Giocatori: quanti sono?
- Fortuna: il caso ha un ruolo?
- Informazione: cosa sanno?
- Ricompensa: qual è l'incentivo?
- Strategia: cosa fare?
- Razionalità: come pensano?

Razionalità dei giocatori

Si assume che i giocatori vogliano massimizzare la ricompensa totale (il *valore* finale del gioco).

Poker



- Quattro giocatori.
- Ci sono le carte (gioco *non deterministico*).
- Informazione *imperfetta*: non conosciamo la mano dell'avversario.
- Si vince una somma in denaro, pagata dall'avversario (gioco a *somma zero*).
- Strategia: bluff/non bluff.
- Si cerca di vincere il più possibile (*razionalità*).

Gioco in borsa



- Giocatori: circa 10^8 .
- Non deterministico: si basa su notizie non verificabili.
- Informazione imperfetta.
- Ricompensa: somma zero.
- Le mosse possibili sono le stesse per tutti i giocatori (gioco *simmetrico*).
- Gioco *simultaneo* (le transazioni sono fatte in tempi diversi, ma senza sapere le mosse degli altri giocatori).

Trattativa per un salario

- Due giocatori: futuro impiegato e datore di lavoro.
- Informazione imperfetta: non sappiamo se il datore di lavoro ha altre richieste.
- Ricompensa: lo stipendio (somma zero se non consideriamo le tasse).
- Strategia: rifiutare la prima offerta.

A fari accesi nella notte

- Due giocatori, che guidano su una strada non illuminata in senso opposto.
- Informazione perfetta: vediamo se l'altro guidatore ha acceso o no gli abbaglianti.
- Ricompensa: sicurezza, con gli abbaglianti vediamo meglio la strada.
- Strategia: finché l'altro non lampeggia, li tengo accesi.
- Non simmetrico: non ho controllo sui fari dell'altra macchina.

Asta in busta chiusa per pozzo di petrolio

- Molti giocatori, i partecipanti all'asta.
- Informazione imperfetta: non conosciamo le altre offerte, e neanche i risultati delle loro ricerche.
- Ricompensa: incerta, perché il valore del giacimento si può solo stimare.
- Strategia: investiamo dei soldi nell'analisi del giacimento/corrompiamo qualcuno per farci dire l'offerta dell'avversario.

Altri esempi

- Biologia: competizioni tra animali (John Nash che osserva i piccioni nel film Beautiful Mind).
- Commercio: dove aprire un bar.
- Ingegneria civile: gestione semaforo rosso/verde.
- Scienze comportamentali: provare a sedersi in una metropolitana affollata.
- Guerra: spesso modellata con il *chicken game*.
- Sistemi di voto, quando sposarsi, donazione/trapianto di reni, associazione neolaureati in medicina/ospedale preferito. . .

E il Go?



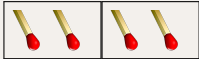
Giochi combinatori

Il Go, gli scacchi, la dama, il tris, il Nim, e tanti altri, sono *giochi combinatori*:

- Giocatori: 2.
- Fortuna: no.
- Informazione: perfetta. Si gioca uno dopo l'altro, e si vedono le mosse dell'avversario.
- Ricompensa: vittoria/sconfitta (+1/-1).

Un gioco molto semplice: il Nim

Regole del Nim

- Due gruppetti di fiammiferi: 
- Ogni giocatore, a turno, toglie uno o più fiammiferi (anche tutti) da un gruppetto a scelta.
- Perde chi non può togliere nessun fiammifero (vince chi toglie l'ultimo fiammifero).

Nim: partita di esempio

- Due giocatori: Zio Massimo e Zio Minimo.

- Situazione iniziale: 

- Zio Massimo toglie un fiammifero dal gruppetto di sinistra:



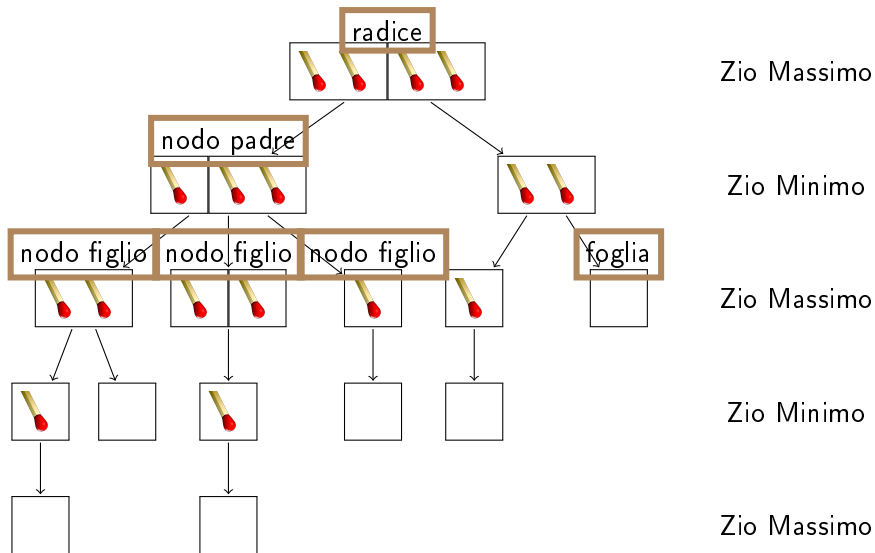
- Zio Minimo toglie un fiammifero dal gruppetto di destra:



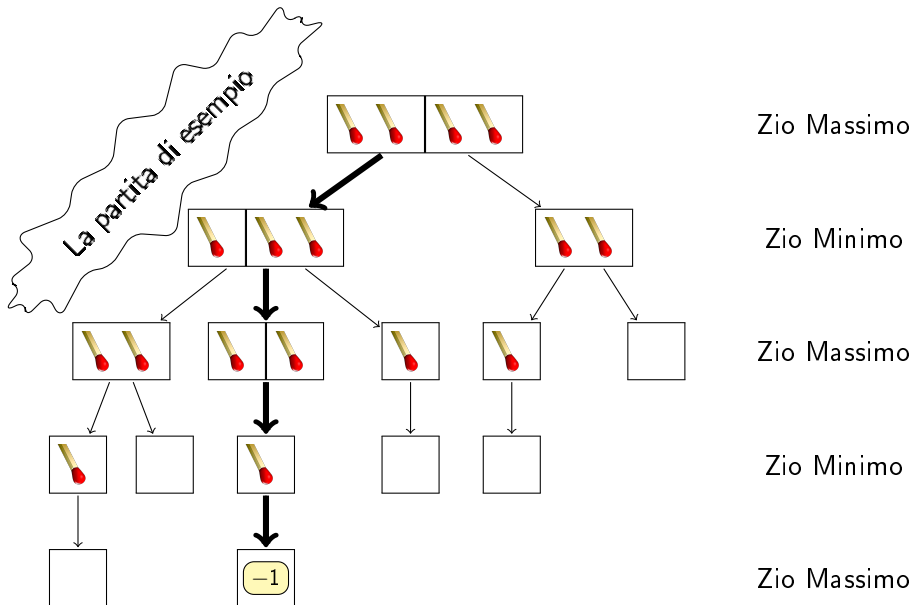
- Zio Massimo toglie un fiammifero dal gruppetto di sinistra - deve farlo, non può passare: 

- Zio Minimo toglie l'ultimo fiammifero e vince: 

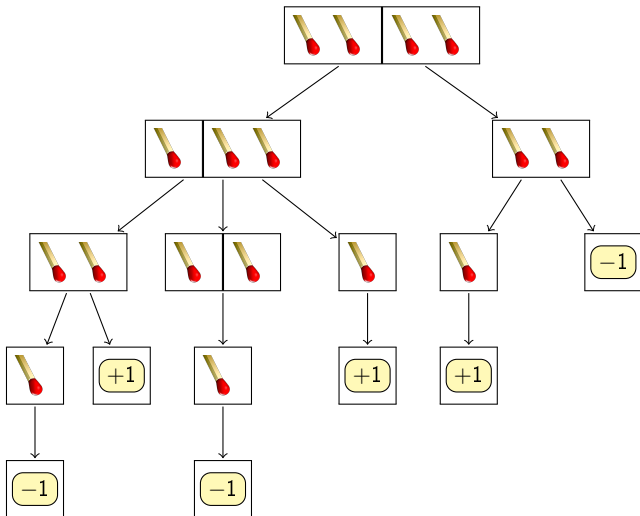
Nim: albero delle partite



Nim: albero delle partite



Nim: albero delle partite



Zio Massimo

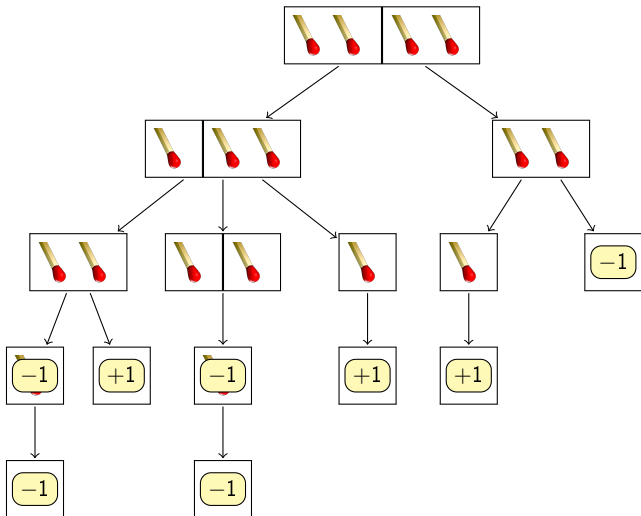
Zio Minimo

Zio Massimo

Zio Minimo

Zio Massimo

Nim: albero delle partite



Zio Massimo

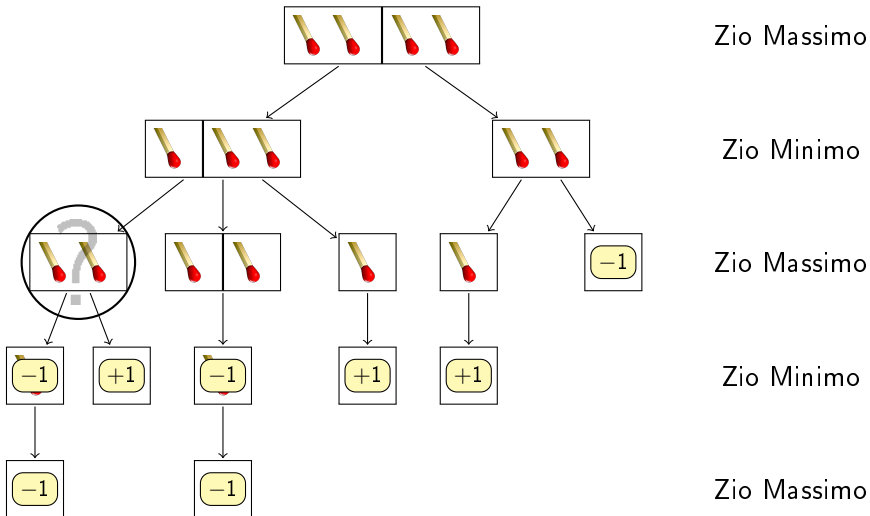
Zio Minimo

Zio Massimo

Zio Minimo

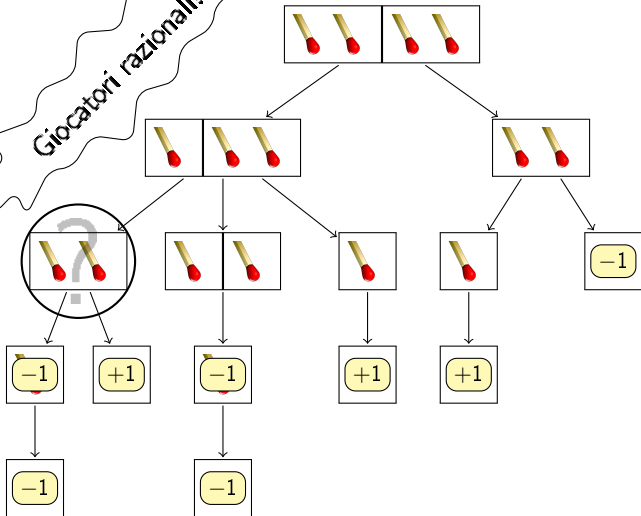
Zio Massimo

Nim: albero delle partite



Nim: albero delle partite

Giocatori razionali!!



Zio Massimo

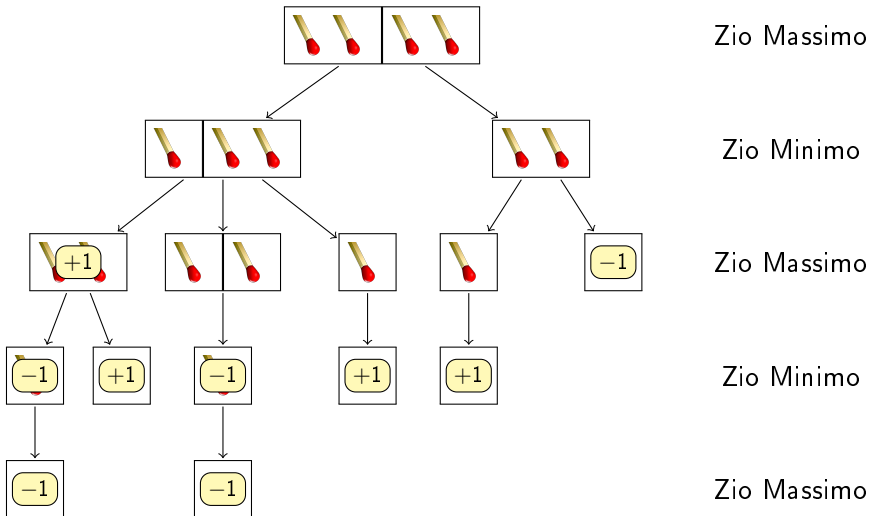
Zio Minimo

Zio Massimo

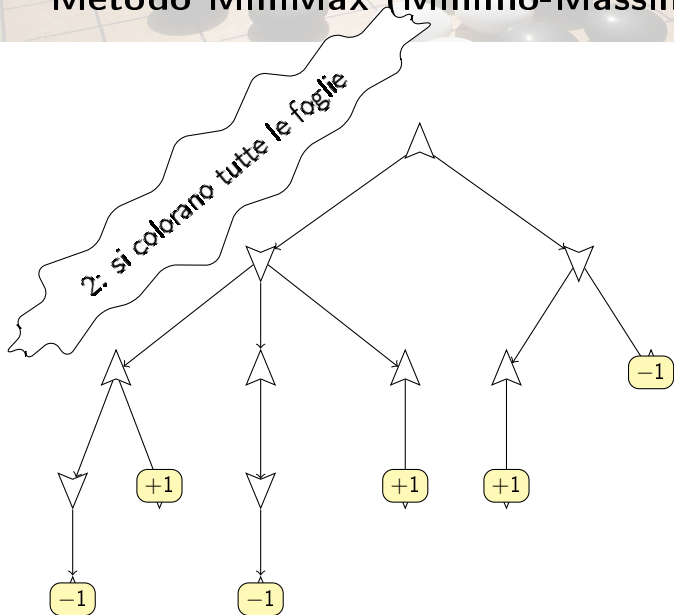
Zio Minimo

Zio Massimo

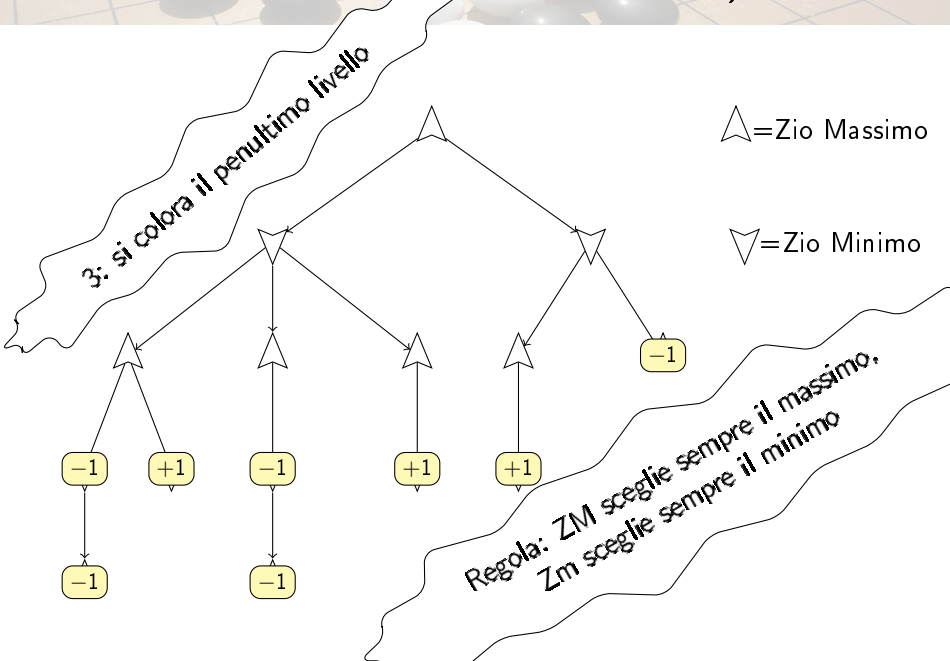
Nim: albero delle partite



Metodo MiniMax (Minimo-Massimo)

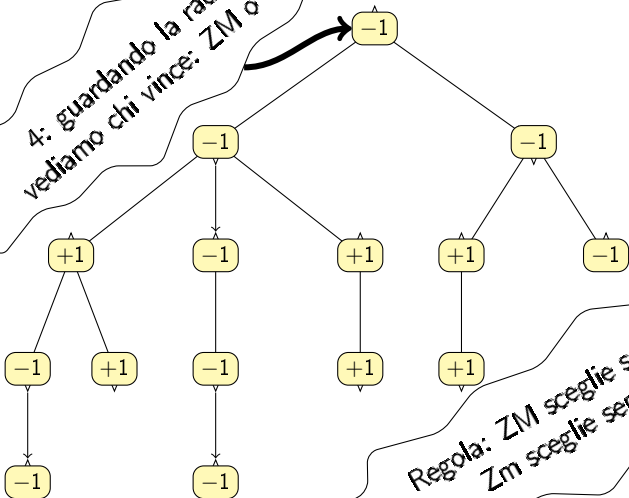


Metodo MiniMax (Minimo-Massimo)



Metodo MiniMax (Minimo-Massimo)

4: guardando la radice, vediamo chi vince: ZM o Zm



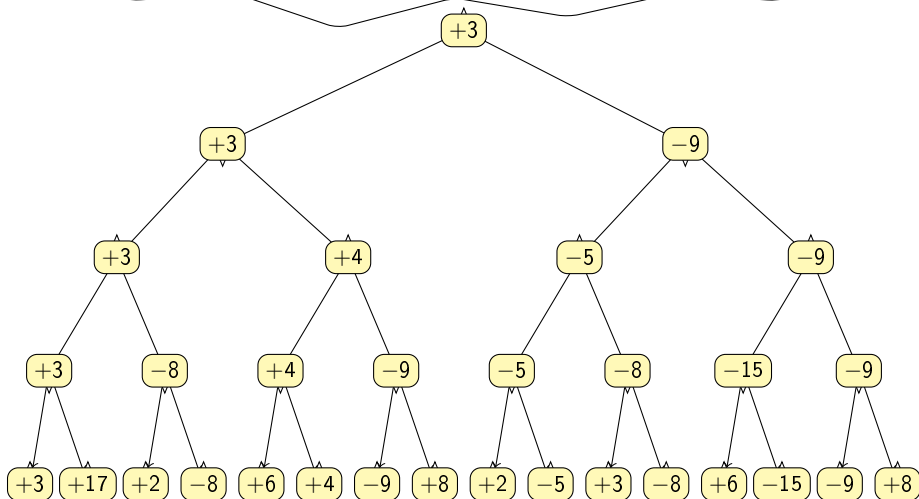
△ = Zio Massimo

▽ = Zio Minimo

Regola: ZM sceglie sempre il massimo,
Zm sceglie sempre il minimo

Risolvere un gioco

La colorazione dell'albero ha una conseguenza importante: a partire da qualunque posizione, possiamo giocare la *partita perfetta*



Risolvere un gioco

Chomp, il gioco del cioccolato avvelenato

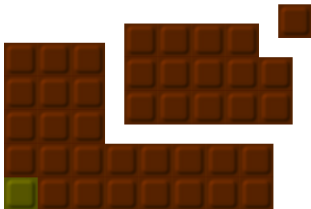
Un gioco si può risolvere anche in maniera *non costruttiva*: si sa che chi comincia vince (o perde), ma non si conosce una strategia



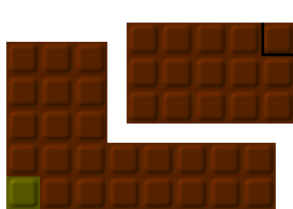
(a)



(b)



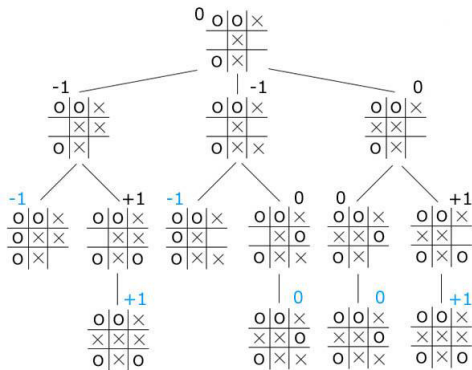
(c)



(d)

Giochi combinatori finiti

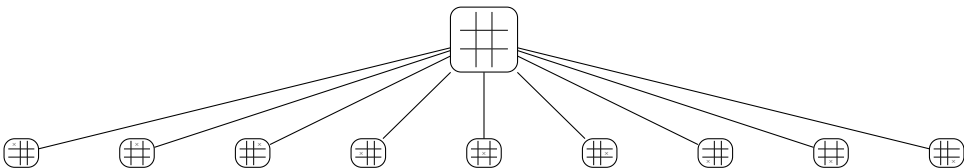
Tutti i giochi "tipo Nim" si possono rappresentare con un albero...



... e quindi risolvere con il MiniMax!

Giochi combinatori finiti

Problema: l'albero cresce molto velocemente, e il MiniMax esamina tutte le foglie!



9 · 8 nodi



9 · 8 · 7 nodi

Foglie=9! = 9 · 8 · 7 · 6 · 5 · 4 · 3 · 2 = 362880

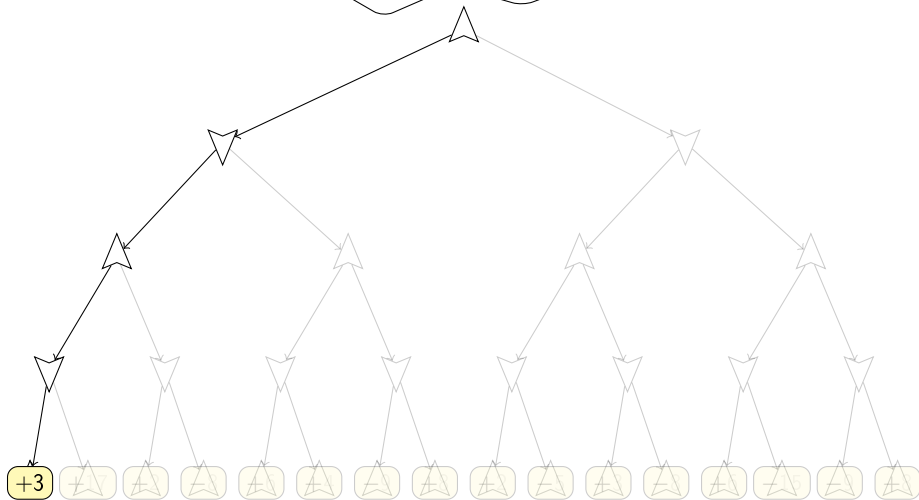
Potatura Alfa-Beta

Serve un modo per navigare l'albero il meno possibile!

- Idea: tagliare i “rami secchi”.
- “Se io gioco qui, faccio minimo 3 punti. Se invece gioco qui, il mio avversario ha una mossa che mi fa fare massimo 2 punti. Quindi questa seconda opzione non la considero.”
- La potatura Alfa-Beta è un algoritmo che formalizza questa idea di “faccio minimo 3 punti” e “faccio massimo 2 punti”.

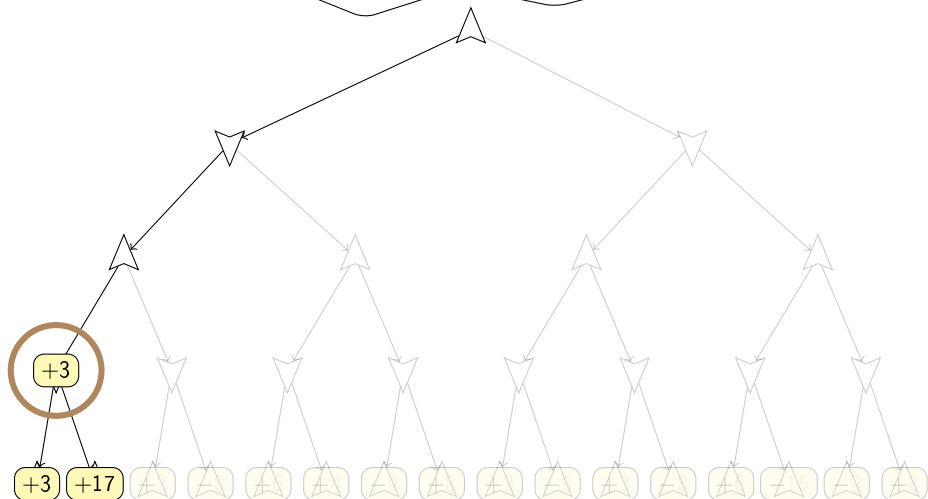
Potatura Alfa-Beta

Consideriamo per ora solo questo ramo



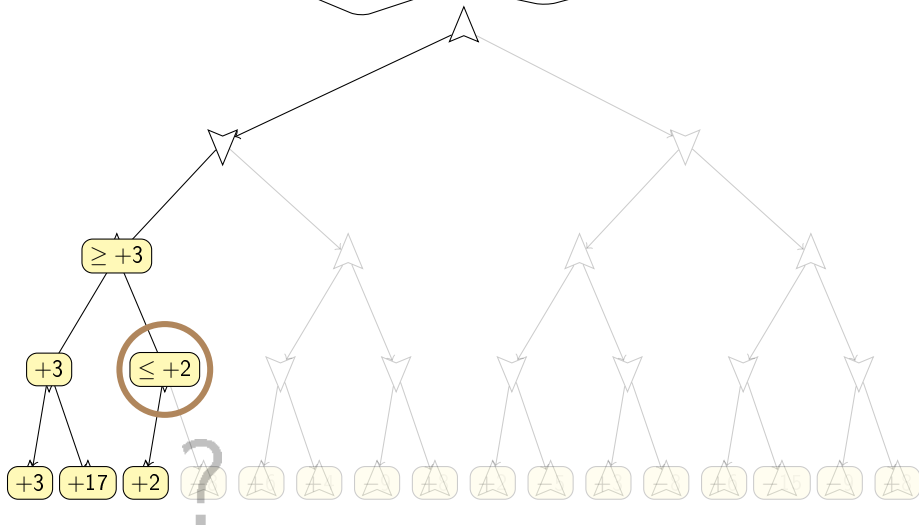
Potatura Alfa-Beta

Il figlio successivo è +17, quindi Zio Minimo sceglie +3



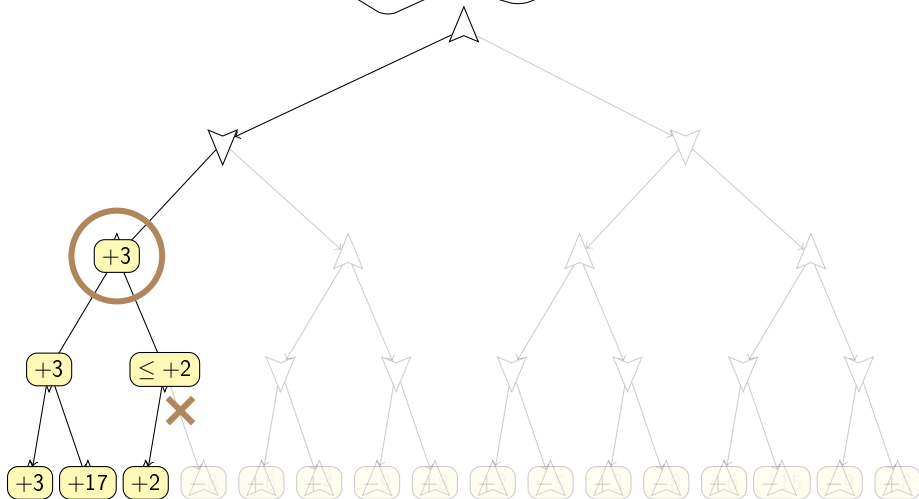
Potatura Alfa-Beta

Zio Minimo sceglierà il minimo tra +2 e qualcos'altro



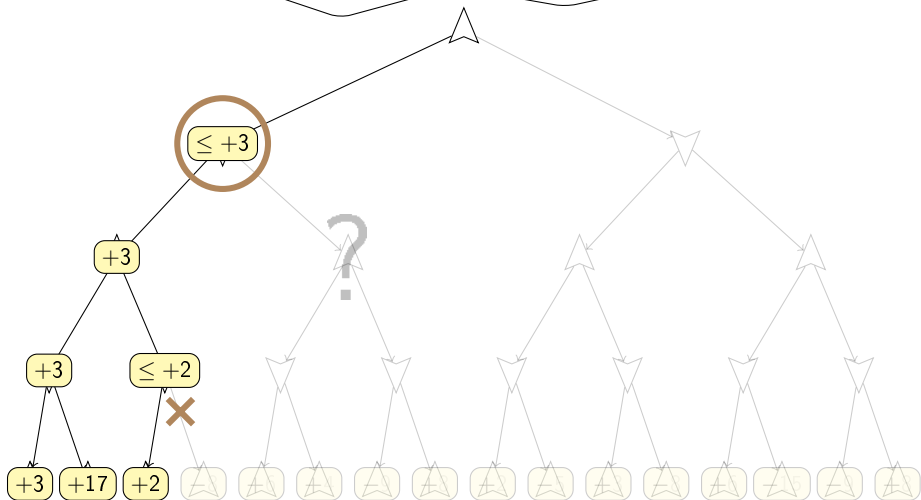
Potatura Alfa-Beta

Abbiamo potato una parte di albero!



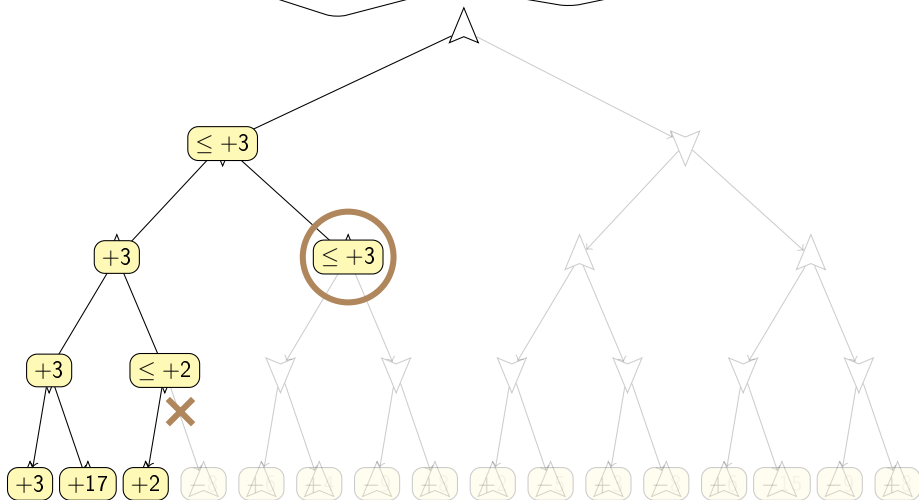
Potatura Alfa-Beta

Continuiamo così, passando ai nodi le informazioni che abbiamo...



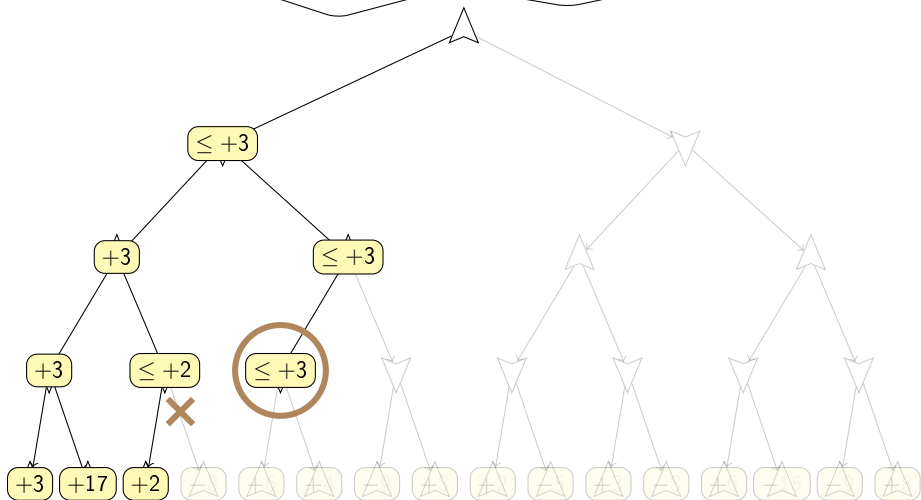
Potatura Alfa-Beta

Se otteniamo > 3 , vuol dire che nel nodo padre possiamo mettere 3!



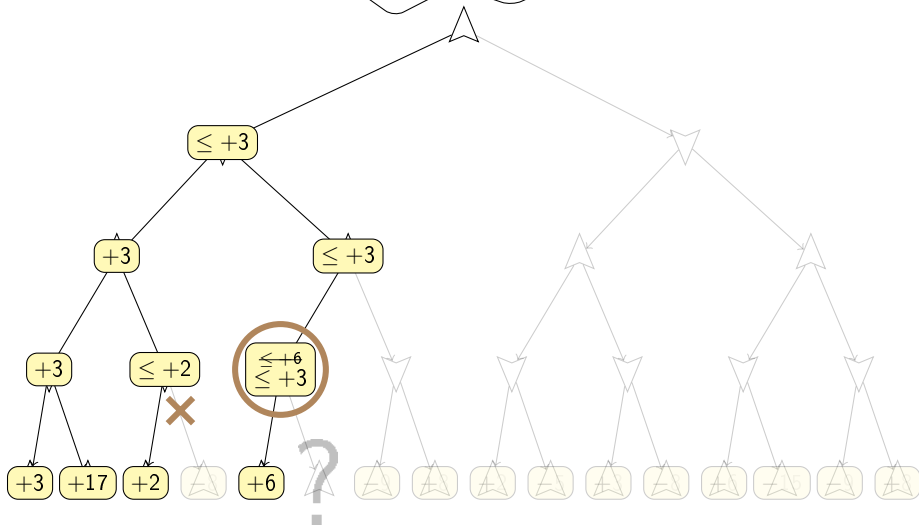
Potatura Alfa-Beta

Se otteniamo > 3 , vuol dire che nel nodo padre c'è un numero > 3 !



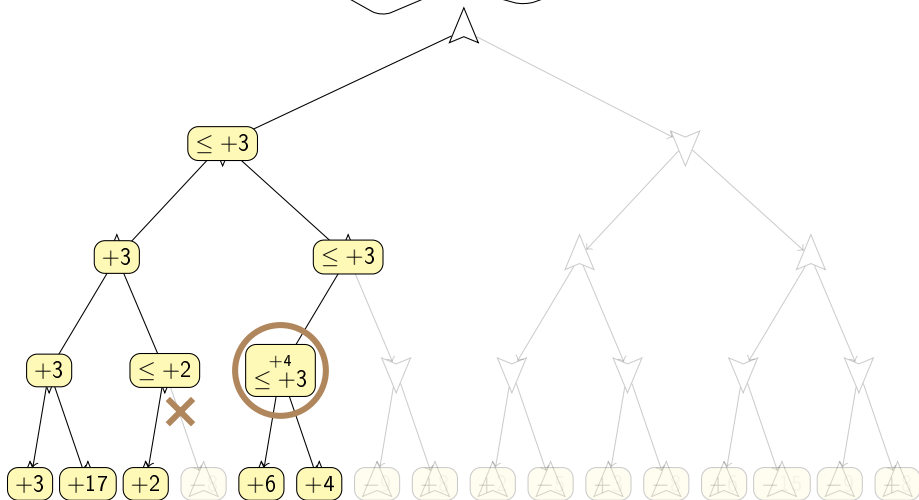
Potatura Alfa-Beta

≤ 6 non serve, abbiamo già ≤ 3



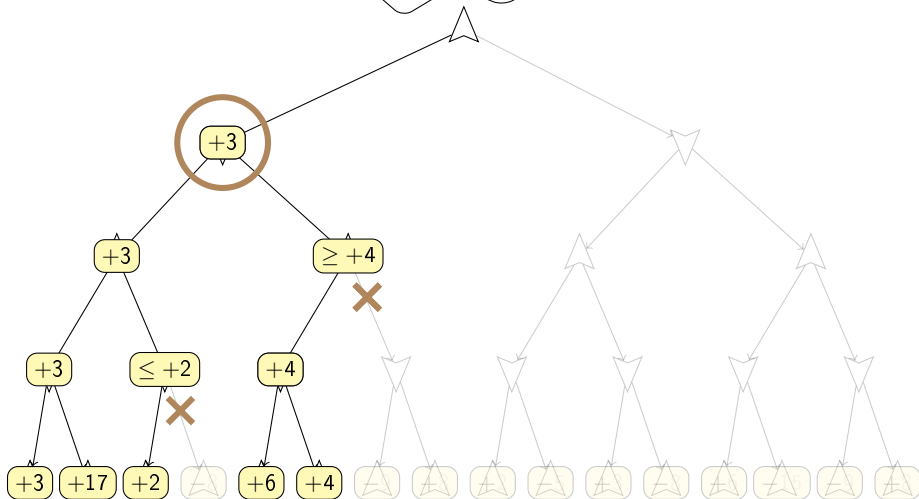
Potatura Alfa-Beta

Abbiamo trovato 4, ma ci serviva ≤ 3 ...



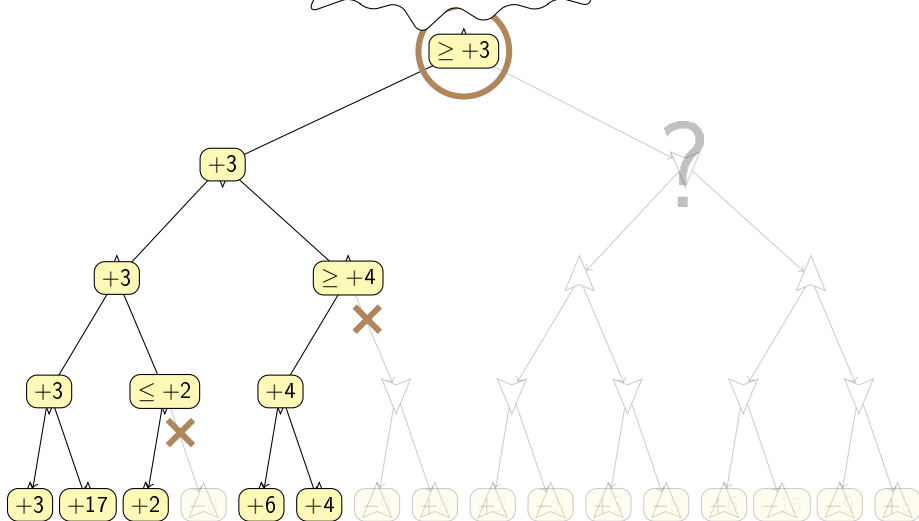
Potatura Alfa-Beta

...quindi potiamo ancora!



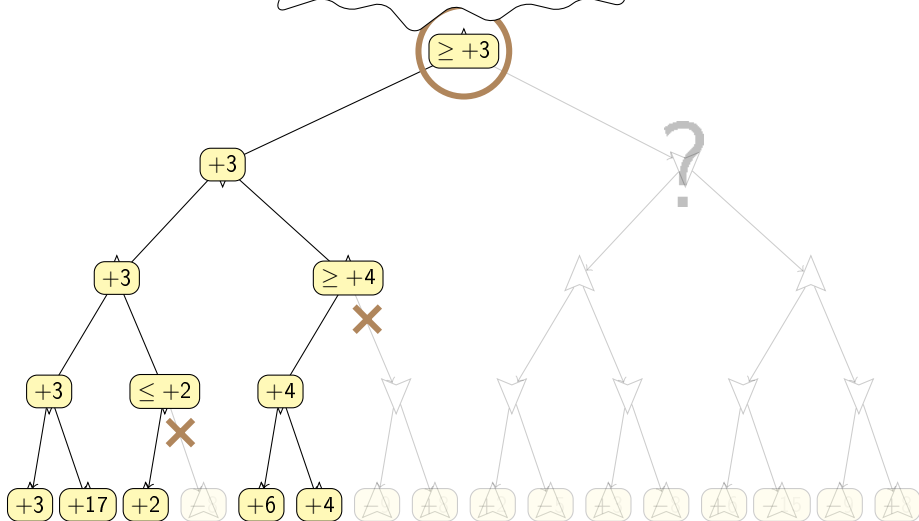
Potatura Alfa-Beta

Siamo in cima!



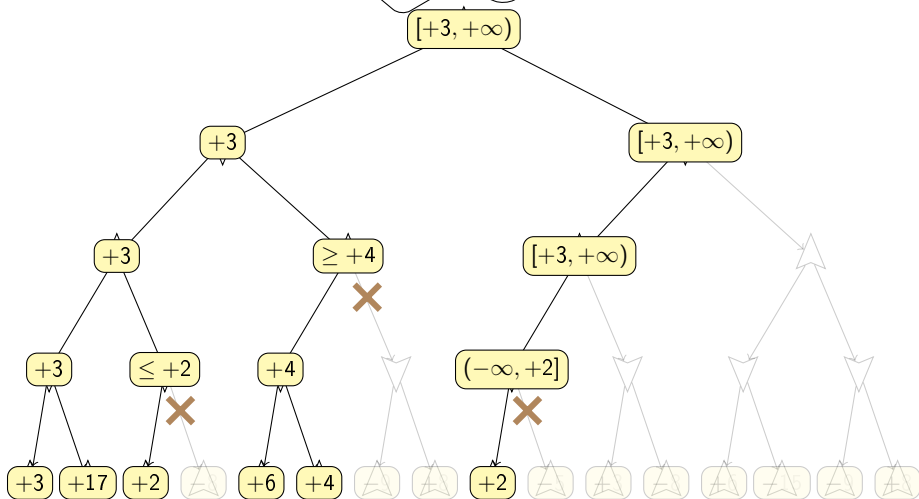
Potatura Alfa-Beta

Perché "Alfa-Beta"?



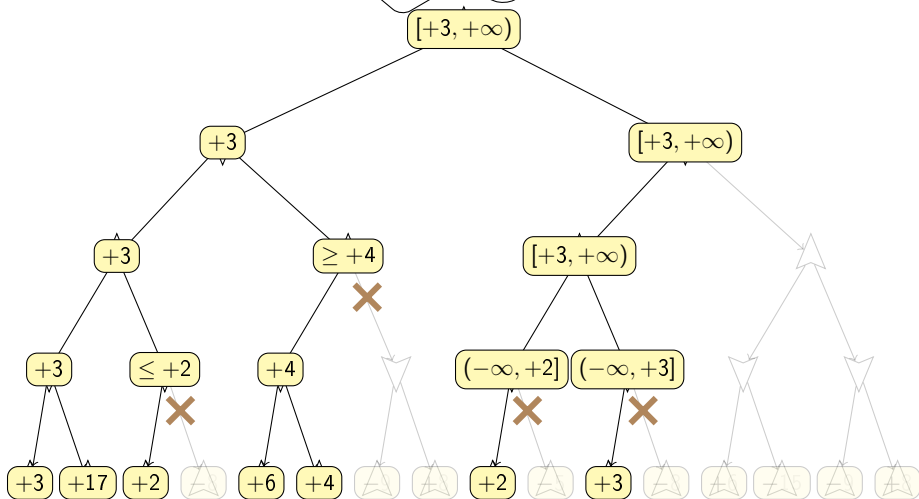
Potatura Alfa-Beta

Proseguiamo come prima...



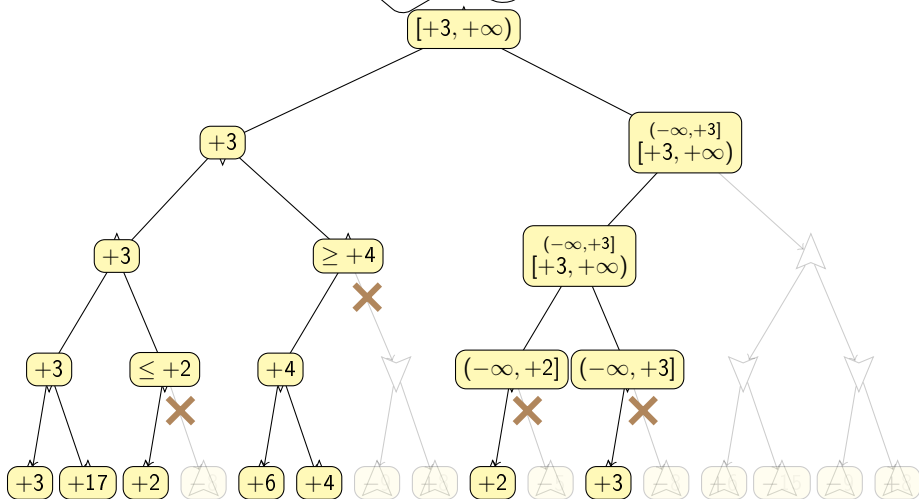
Potatura Alfa-Beta

Proseguiamo come prima...



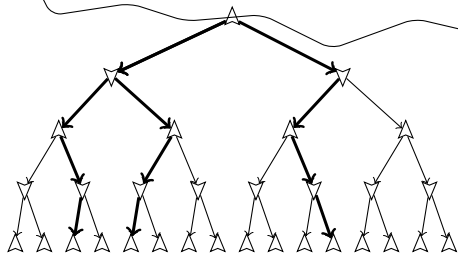
Potatura Alfa-Beta

Proseguiamo come prima...



Complessità dei giochi

La complessità dell'albero dipende da quante possibilità ci sono a ogni mossa, in media



2 nodi

$2 \cdot 2 = 4$ nodi

$2 \cdot 2 \cdot 2 = 8$ nodi

$2 \cdot 2 \cdot 2 \cdot 2 = 16$ foglie/cammini

Molti giochi hanno un elevato *fattore di ramificazione*

Dama: $r = 10$, 20
livelli: 10^{20} foglie!

Scacchi: $r = 20$, 40 livelli: $20^{40} =$
 $1,099511628 \cdot 10^{52}$ foglie!

Complessità dei giochi

Othello: 10^{58} foglie. Non risolto, ma il computer batte gli umani dal 1980.

Dama: 10^{20} foglie. *Chinook* comincia i calcoli nel 1989: dopo 10^{14} calcoli in 18 anni, nel 2007 risolve il gioco.

- Mancala. Risolto con l'albero, pareggio.
- Forza 4. Risolto con l'albero, chi comincia vince.
- Hex. Risolto con il furto, chi comincia vince.
- Nim. Risolto con l'albero, dipende dal numero di colonne e di pezzi in ogni colonna.
- Tris. Risolto con l'albero, pareggio.

Scacchi: 10^{52} foglie. *Deep Blue* sconfigge il campione del mondo nel 1997. 200 milioni di posizioni al secondo, 40 livelli di profondità. Non risolto.

Parliamo di Go... Finalmente!

- È un gioco di strategia astratto: non ci sono dadi, non c'è il caso, il risultato è tutto colpa - e merito - nostro!
- È un gioco antichissimo, come si evince dal ben noto detto: "食日、所用心、矣哉、不有博弈者乎、之乎已。"
- Regole del Go: semplicissime.
- Strategia nel Go: complicatissima, anche per i computer.
Infatti:

Il miglior programma di Go non gioca meglio di un forte dilettante.


Parliamo di Go... Finalmente!

- È un gioco di strategia astratto: non ci sono dadi, non c'è il caso, il risultato è tutto colpa - e merito - nostro!
- È un gioco antichissimo, come si evince dal ben noto detto: "食日、所用心、矣哉、不有博弈者乎、之 乎已。"
- Regole del Go: semplicissime.
- Strategia nel Go: complicatissima, anche per i computer.
Infatti:

Il miglior programma di Go non gioca meglio di un forte dilettante.



AlphaGo



- Fan Hui: giocatore di Go professionista, livello 2 dan, campione europeo dal 2013.
- AlphaGo: software sviluppato da Google.
- Ottobre 2015: AlphaGo batte Fan Hui 5-0 in un torneo ufficiale.

28 gennaio 2016

Esce la descrizione del software sulla rivista Nature.

Aprire `alphago.mp4`

AlphaGo

Teoria dei giochi combinatoria

Regole del Go

Metodo Montecarlo per la ricerca di alberi

Reti neurali

Apprendimento automatico

Alberi, MiniMax e potatura Alfa-Beta

AlphaGo

Teoria dei giochi, giocatore di Go professionista, livello 2 dan.

completato

Regole del Go

ottobre 2015: AlphaGo
ciale.

Metodo Montecarlo per la ricerca di alberi

Reti neurali

Apprendimento automatico


ennaio

ce la descrizione del software sulla rivista Science.

Alberi, Minimax e ricerca Alpha-Beta

completato

Regole del Go

A photograph of several Go stones (black and white) on a wooden board, positioned in the top right corner of the slide.

Aprire `zio-massimo-vs-zio-minimo.sgf`

Aprire `fanhui-alphago.xspf`

Aprire `fanhui-alphago.sgf`

Complessità nel Go

Nel Go abbiamo circa 200 mosse possibili a ogni turno, e una media di 150 turni a partita: $200^{150} \cong 10^{345}$

Anche con Alfa-Beta, se va bene abbiamo 100 mosse invece di 200

Non possiamo permetterci di esaminare tutto l'albero, possiamo esaminare solo pochi nodi a ogni mossa.

Atomi nell'universo: 10^{80}

Particelle nell'universo: 10^{87}

Complessità nel Go

Nel Go abbiamo circa 200 mosse possibili a ogni turno, e una media di 150 turni a partita: $200^{150} \cong 10^{345}$

Anche con Alfa Beta, se va bene abbiamo 100 mosse invece di 200

Primo turno nel Go

Non possiamo permetterci di esaminare tutto l'albero, possiamo esaminare solo pochi nodi a ogni mossa.

Atomi nell'universo: 10^{80}

Particelle nell'universo: 10^{87}

Complessità nel Go

Nel Go abbiamo circa 200 mosse possibili a ogni turno,
una media di 150 turni a partita: $200^{150} \cong 10^{345}$

Il metodo Montecarlo
serve a scegliere i nodi
da esaminare e i nodi
da scartare

Primo turno nel Go

Non possiamo permetterci di esaminare tutto l'albero,
possiamo esaminare solo pochi nodi a ogni mossa

Atomi nell'universo: 10^{80}

Particelle nell'universo: 10^{87}

AlphaGo

Tecnica di Go professionista, livello 2 dan.
13.

Completato

Regole del Go

naGo sotto licenza MIT

ottobre 2015: AlphaGo
ciale.

Metodo MonteCarlo per la ricerca di alberi

Reti neurali

Apprendimento automatico

ennaio 2016

ce la descrizione del software sulla rivista Science.

Alberi, Minimax e ricerca Alpha-Beta

Completato

AlphaGo

Tecnica di Go professionista, livello 2 dan.

13.

Completato

ottobre 2015: AlphaGo

ficiale.

Metodo MonteCarlo per la ricerca di alberi

Reti neurali

Apprendimento automatico

ennaio

ce la descrizione del software sulla rivista

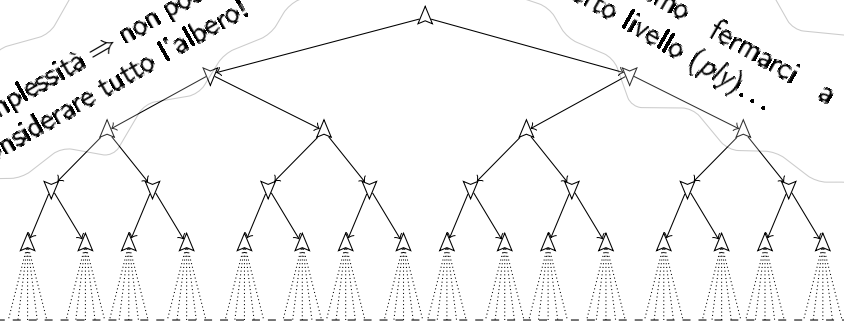
Alberi, Minimax e Monte Carlo Alpha-Beta

Completato

La funzione valore

Complessità \Rightarrow non possiamo considerare tutto l'albero!

Dobbiamo fermarci a un certo livello (ply)...

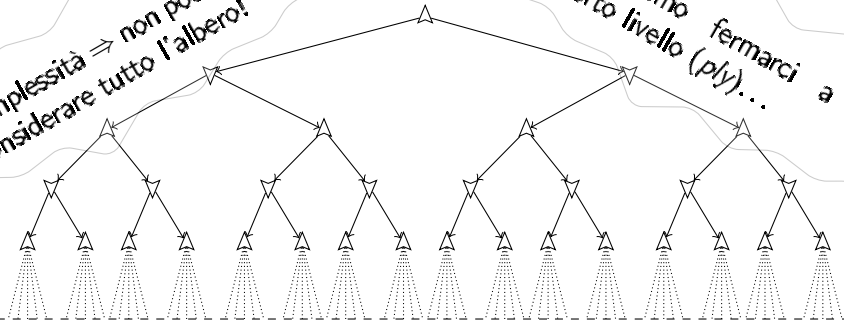


... ma la colorazione dell'albero parte dalle foglie!

La funzione valore

Complessità \Rightarrow non possiamo considerare tutto l'albero!

Dobbiamo fermarci a un certo livello (ply)...



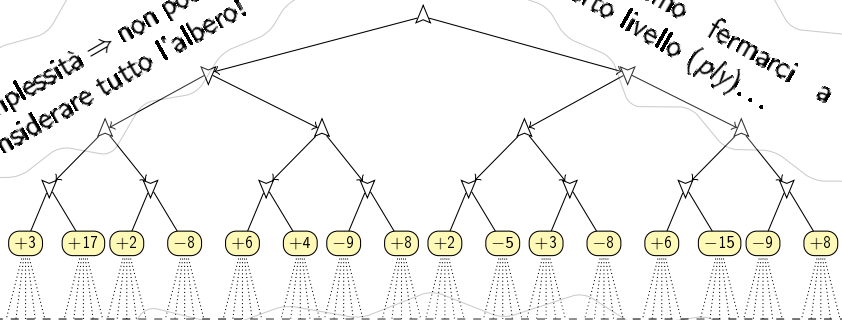
Come facciamo?

... ma la colorazione dell'albero parte dalle foglie!

La funzione valore

Complessità \Rightarrow non possiamo considerare tutto l'albero!

Dobbiamo fermarci a un certo livello (ply)...



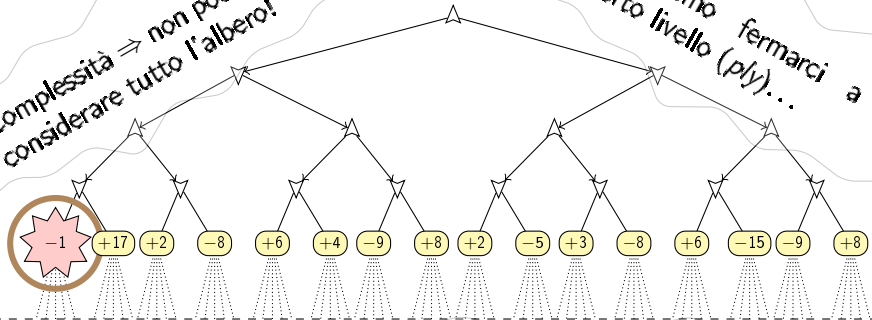
Soluzione: un genio della lampada ci suggerisce i valori da mettere nelle "foglie"!

Questo genio si chiama *funzione valore*

La funzione valore

Complessità \Rightarrow non possiamo considerare tutto l'albero!

Dobbiamo fermarci a un certo livello (ply)...



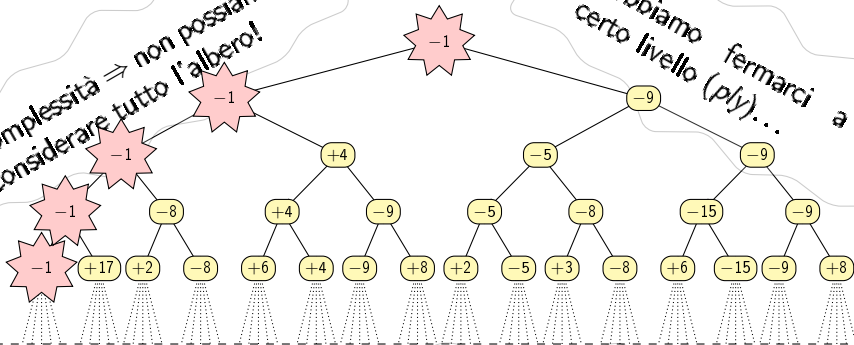
Se la funzione valore è imprecisa, il risultato può non essere corretto...

Questo genio si chiama *funzione valore*

La funzione valore

Complessità \Rightarrow non possiamo considerare tutto l'albero!


Dobbiamo fermarci a un certo livello (ply)...



... e un vincitore può trasformarsi in perdente!

Questo genio si chiama *funzione valore*

La funzione valore

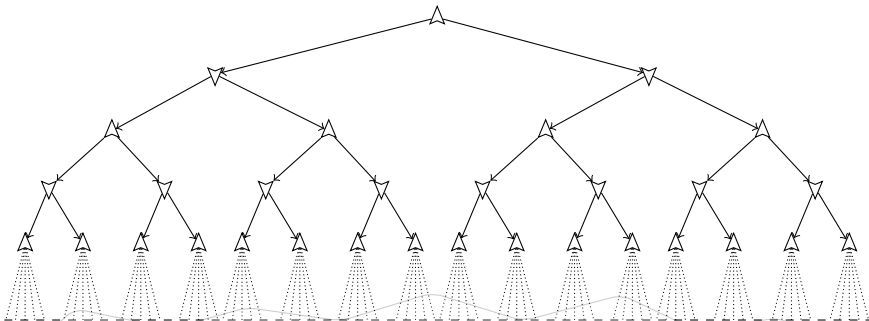


- Ogni volta che in un gioco pensiamo “stiamo vincendo” o “stiamo perdendo”, stiamo in realtà usando una “nostra” funzione valore!
- Chi gioca a dama, scacchi o Go sa valutare a occhio se la partita sta andando male o bene: usa una sua personale funzione valore!
- Farlo fare a un computer non è semplice: nel Go era l’ostacolo principale!

Esempi semplici di funzione valore

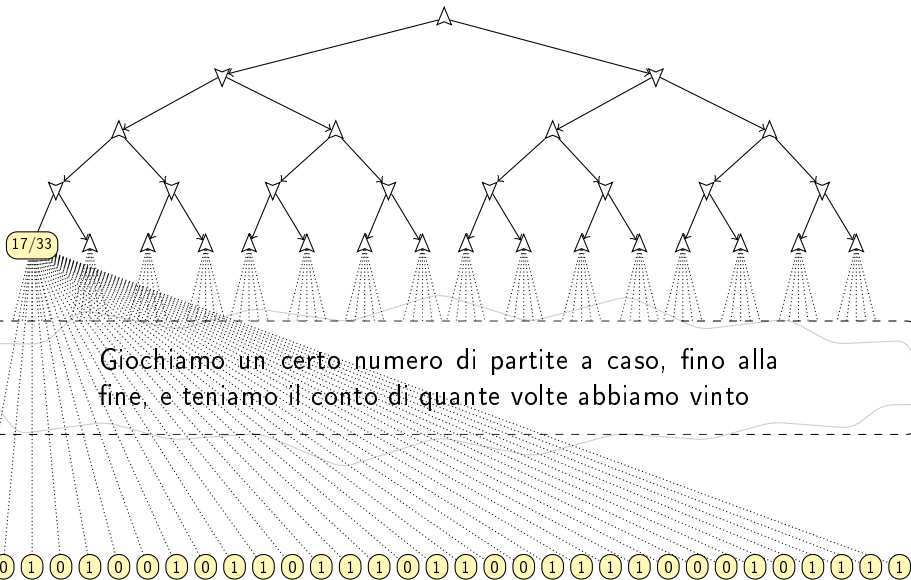
- Nella dama, una funzione valore assegna alle pedine 1, alle dame 2, e il valore del pezzo viene moltiplicato per la distanza dalla linea di fondo al quadrato.
- Negli scacchi, è un numero ottenuto dai pezzi in gioco: pedoni 1, alfieri e cavalli 3, torri 5, regine 10.
- Deep Blue usa una “somma pesata” di varie caratteristiche: $w_1 \cdot C_1 + \dots + w_n \cdot C_n$. Le caratteristiche sono circa 6000, tra cui la posizione dei singoli pezzi e le configurazioni di più pezzi insieme.

Una funzione valore "a caso"

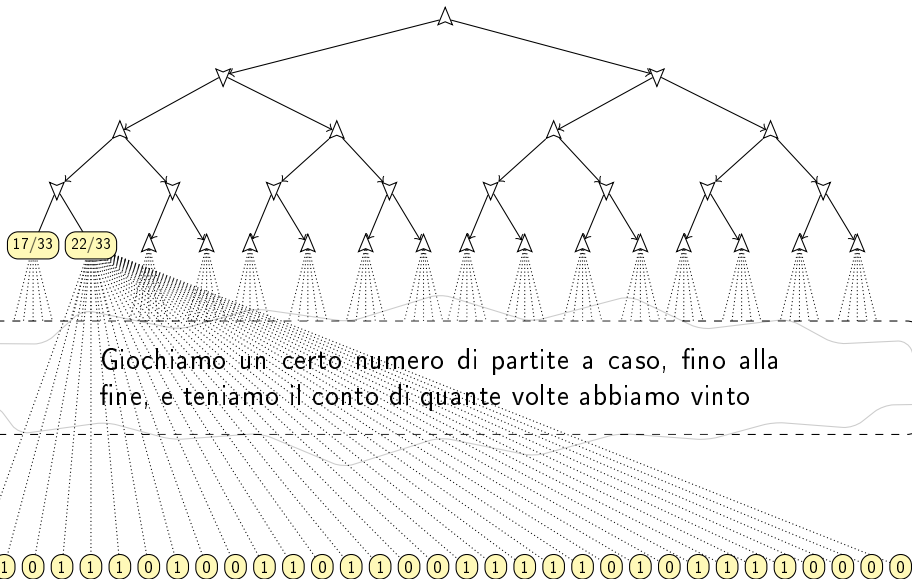


Giochiamo un certo numero di partite a caso, fino alla fine, e teniamo il conto di quante volte abbiamo vinto

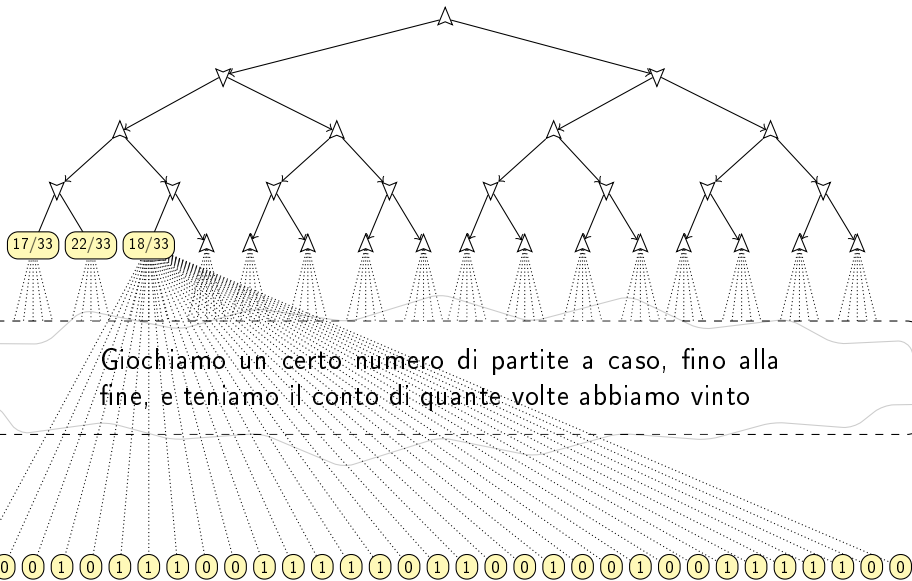
Una funzione valore "a caso"



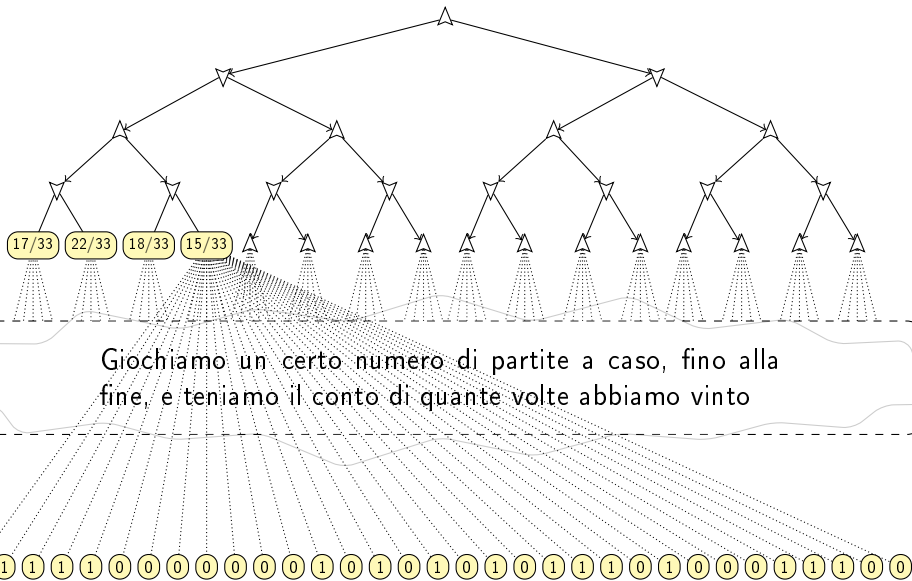
Una funzione valore "a caso"



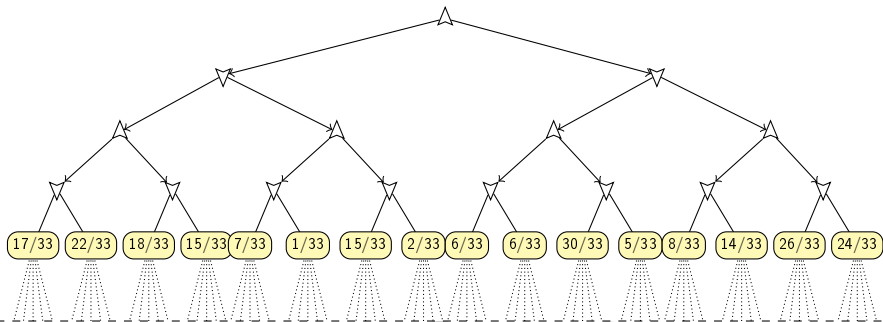
Una funzione valore "a caso"



Una funzione valore "a caso"

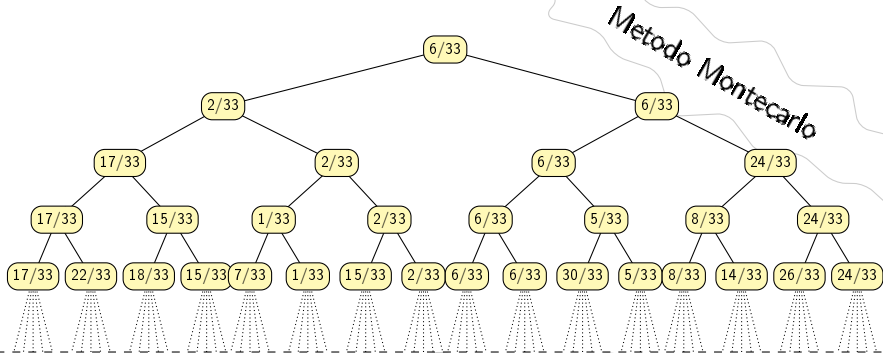


Una funzione valore "a caso"



E così via, fino ad avere un valore in tutte le "foglie"

Una funzione valore "a caso"



E così via, fino ad avere un valore in tutte le "foglie"

Ora possiamo colorare l'albero fino alla radice!

Metodo Montecarlo: che cos'è?

1/6



Con un esperimento identico e ripetuto molte volte valuto una quantità che non conosco.

1/6



1/6



Voglio vedere se un dado è truccato: come faccio? Se non fosse truccato, dovrebbe darmi 1, 2, 3, 4, 5, 6 con le stesse probabilità.

1/6



1/6



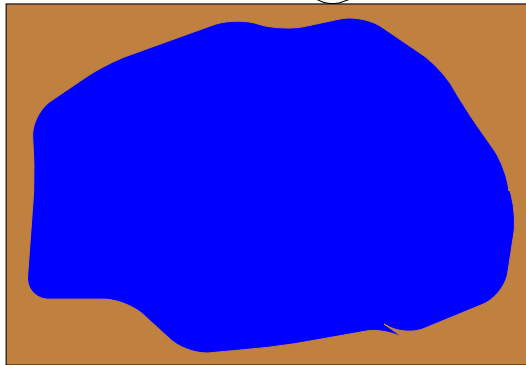
Lo lancio 60 volte, e conto quante volte esce ciascuna faccia.

1/6



Metodo Montecarlo: che cos'è?

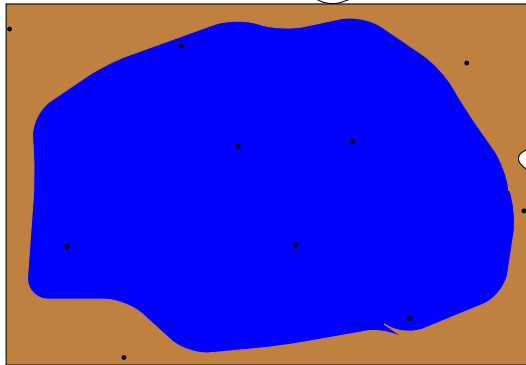
Problema: voglio misurare la superficie di un lago.



Se ho un esercito a disposizione, uso il metodo delle palle di cannone.

Metodo Montecarlo: che cos'è?

Problema: voglio misurare la superficie di un lago.



Colpi: 10

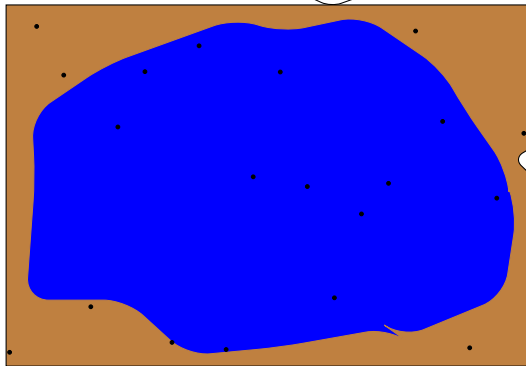
In acqua: 6

$$\text{Stima: } \frac{6}{10} \cdot 7 \cdot 5 = 21 \text{ km}^2$$

Se ho un esercito a disposizione, uso il metodo delle palle di cannone.

Metodo Montecarlo: che cos'è?

Problema: voglio misurare la superficie di un lago.



Colpi: 20

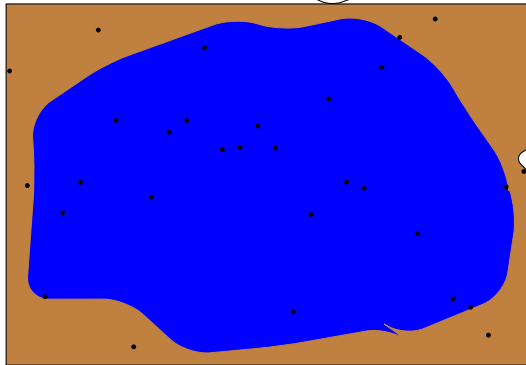
In acqua: 8

Stima: $\frac{12}{20} \cdot 7 \cdot 5 =$
 21 km^2

Se ho un esercito a disposizione, uso il metodo delle palle di cannone.

Metodo Montecarlo: che cos'è?

Problema: voglio misurare la superficie di un lago.



Colpi: 10

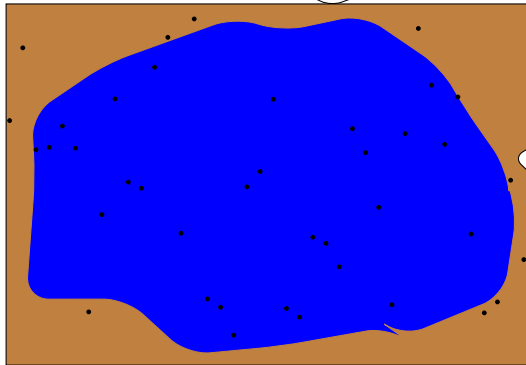
In acqua: 6

$$\text{Stima: } \frac{21}{30} \cdot 7 \cdot 5 = 24.5 \text{ km}^2$$

Se ho un esercito a disposizione, uso il metodo delle palle di cannone.

Metodo Montecarlo: che cos'è?

Problema: voglio misurare la superficie di un lago.



Colpi: 10

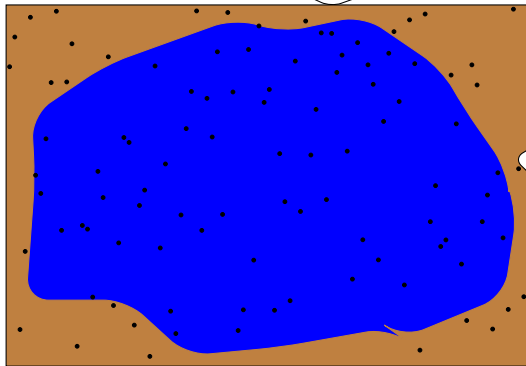
In acqua: 6

$$\text{Stima: } \frac{29}{40} \cdot 7 \cdot 5 = 25.375 \text{ km}^2$$

Se ho un esercito a disposizione, uso il metodo delle palle di cannone.

Metodo Montecarlo: che cos'è?

Problema: voglio misurare la superficie di un lago.



Colpi: 100

In acqua: 89

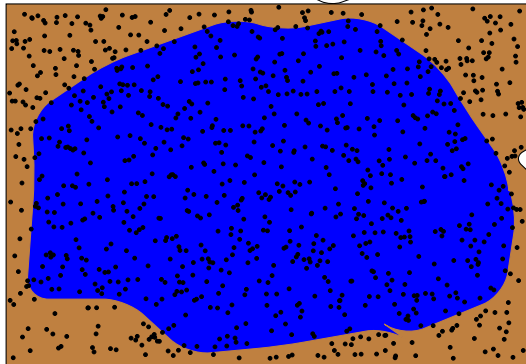
$$\text{Stima: } \frac{89}{100} \cdot 7 \cdot 5 =$$

31.15 km²

Se ho un esercito a disposizione, uso il metodo delle palle di cannone.

Metodo Montecarlo: che cos'è?

Problema: voglio misurare la superficie di un lago.



Colpi: 1000

In acqua: ...


Stima: $\frac{\dots}{1000} \cdot 7 \cdot 5 =$
... km²

Tutto si basa sulla *Legge dei grandi numeri*: più esperimenti faccio, più è precisa la stima!



That's all Folks!

Licenza



- È permesso copiare, distribuire e/o modificare questo documento seguendo i termini della "GNU Free Documentation License", versione 1.2 o ogni versione successiva pubblicata dalla Free Software Foundation; senza sezioni non modificabili, senza testi di prima di copertina e di quarta di copertina. Una copia della licenza è disponibile alla URL:
<http://www.gnu.org/licenses/fdl-1.2-standalone.html>