

Scacchi e Matematica

Paolo Ciancarini
Dipartimento di Scienze dell'Informazione
Università di Bologna

`mailto://ciancarini@cs.unibo.it`
`http://www.cs.unibo.it`

Università di Camerino
23 Maggio 2001

Sommario

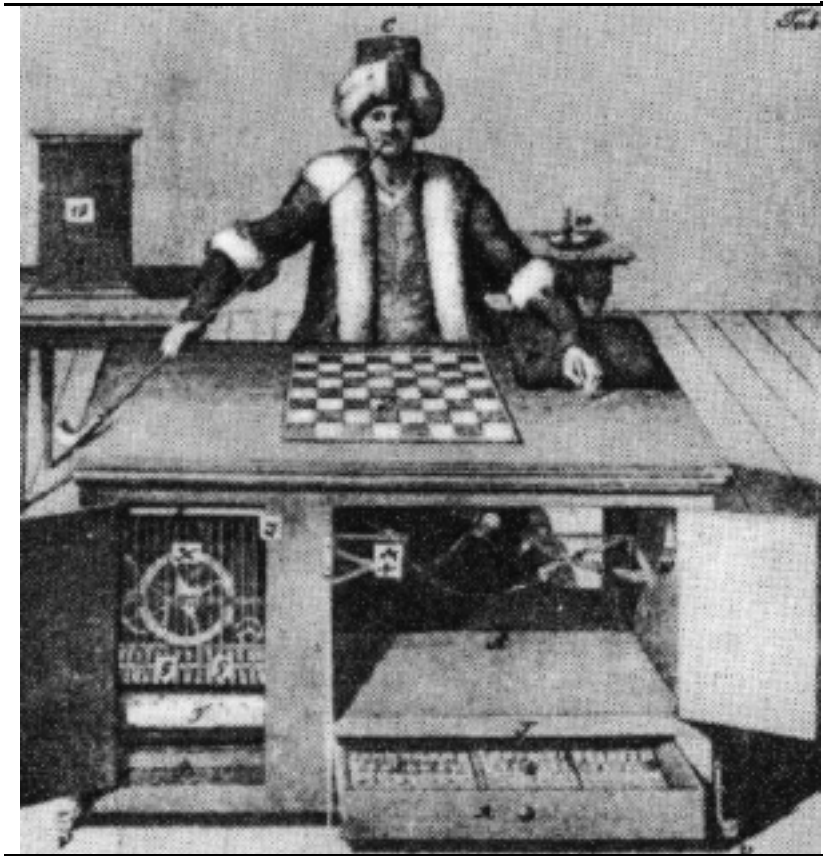
- Perché è interessante studiare i giocatori artificiali
- Storia breve degli automi scacchistici
- Struttura dei giocatori artificiali di scacchi
 - Algoritmi di ricerca su alberi di gioco
 - Funzione di valutazione
 - Valutazione delle prestazioni
- Prospettive future

Evoluzione dei giocatori artificiali

- il problema ingegneristico
- l'approccio teorico-matematico
- il punto di vista cognitivo
- l'approccio agonistico-commerciale

Il primo giocatore artificiale

Il problema: costruire una macchina che gioca (bene) a Scacchi venne formulato per scommessa nel 1769, dal barone Von Kempelen, tecnologo di Maria Teresa d'Austria



Il Turco giocò in molte corti d'Europa e si esibì anche in America; finì bruciato in un incendio a Philadelphia

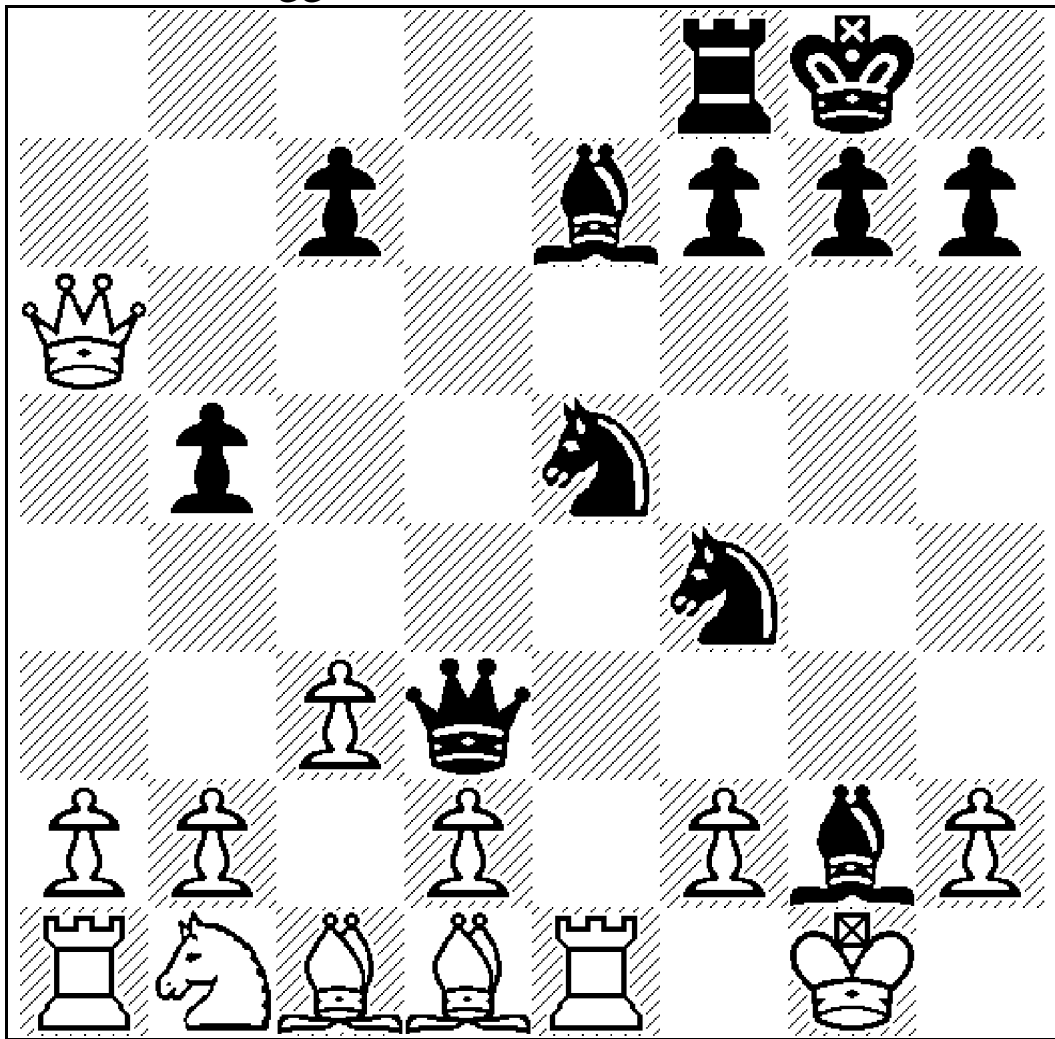
Ha ispirato romanzi e film
Vennero in seguito costruite molte versioni

2001: partita a scacchi nello spazio

Nel film *2001: Odissea nello Spazio* di S. Kubrick, tratto da un libro omonimo di A. Clarke, il computer HAL 9000 gioca a scacchi contro un umano

(HAL sta per Heuristic-Algorithmic Computer, nato ... il 12 gennaio 1997)

Frank-HAL, in viaggio verso Giove, *circa* 2001

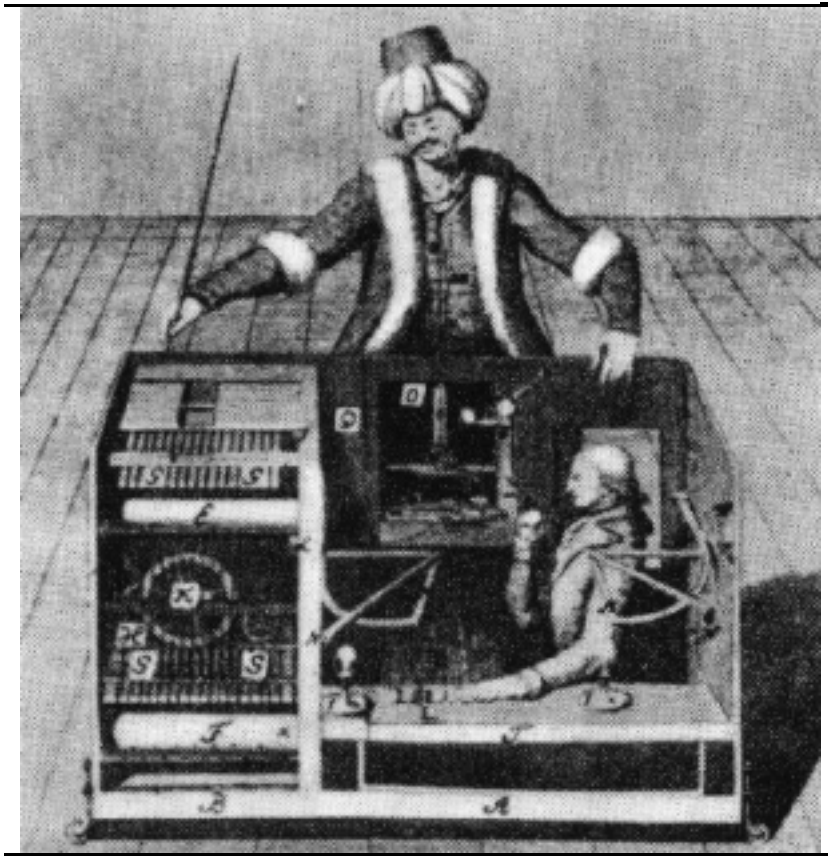


15. ... Df3 e il Bianco abbandona

Edgar Allan Poe e il Turco

“È assolutamente certo che le operazioni dell’Automa sono regolate da una *mente* e da null’altro...”

Se la macchina fosse una pura macchina, l’Automa vincerebbe sempre: invece talvolta è stato sconfitto. Trovato il principio in base al quale si possa ottenere una macchina che gioca a Scacchi, un’estensione di quello stesso principio metterebbe quella macchina in condizione di vincere una partita; un’ulteriore estensione le consentirebbe di vincere tutte le partite.”



In effetti, il Turco era un inganno!
(ma comunque un’opera mirabile di ingegneria)

Cronologia breve

Nel tempo il problema di costruire una vera macchina capace di giocare *bene* a Scacchi è stato affrontato da:

Charles Babbage (~ 1840)

Leonardo Torre y Quevedo (~1890)

Konrad Zuse (~1945)

Claude Shannon (1949)

Alan Turing (1951)

Alan Newell e Herbert Simon (1955-60)

Ken Thompson (1973-85)

e moltissimi altri ...

Charles Babbage e la Macchina Analitica

In un articolo scritto nel 1864 Babbage sosteneva che qualsiasi gioco di scacchiera può essere affrontato con successo da un automa.

Dopo molti studi, ho scelto l'esperimento di costruire una macchina che dovrebbe poter giocare con successo una partita di un gioco di abilità puramente intellettuale, come ad esempio il filetto, la dama, o gli scacchi....

L'automa "esamina" una posizione, e poi comincia a porsi una serie di domande:

1. L'ultima mossa fatta dal mio avversario è legale? Se no, protesto.
2. Ho una posizione indifendibile (ovvero, il matto è inevitabile) ? Se sì, abbandono.
3. Tra quelle possibili, c'è una mossa che mi dà la vittoria (cioè posso dare scacco matto)? Se sì, la dichiaro.
4. L'avversario sta per fare una mossa vincente? Se sì, la prevengo.
5. Se alla prossima mossa non c'è una mossa vincente per uno di noi due, debbo cercare una mossa che crea una doppia minaccia, in modo che il mio avversario ne possa parare una sola; se c'è, la effettuo.
6. Se i primi 5 test falliscono, esamino le mosse successive e in qualche modo ne scelgo una; la effettuo senz'altro.

Le "istruzioni" che compongono tale algoritmo primordiale sono naturalmente piuttosto rozze e imprecise, né Babbage le raffinò ulteriormente. Tanto per cominciare, l'istruzione 6 è tutt'altro che ben definita: che vuol dire precisamente "in qualche modo ne scelgo una"?

La macchina di Torre y Quevedo

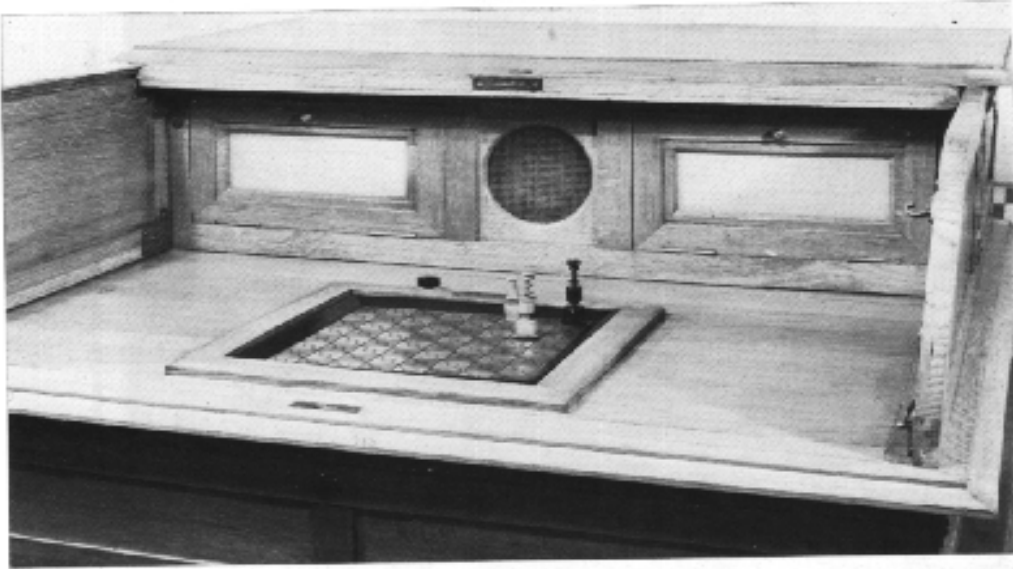
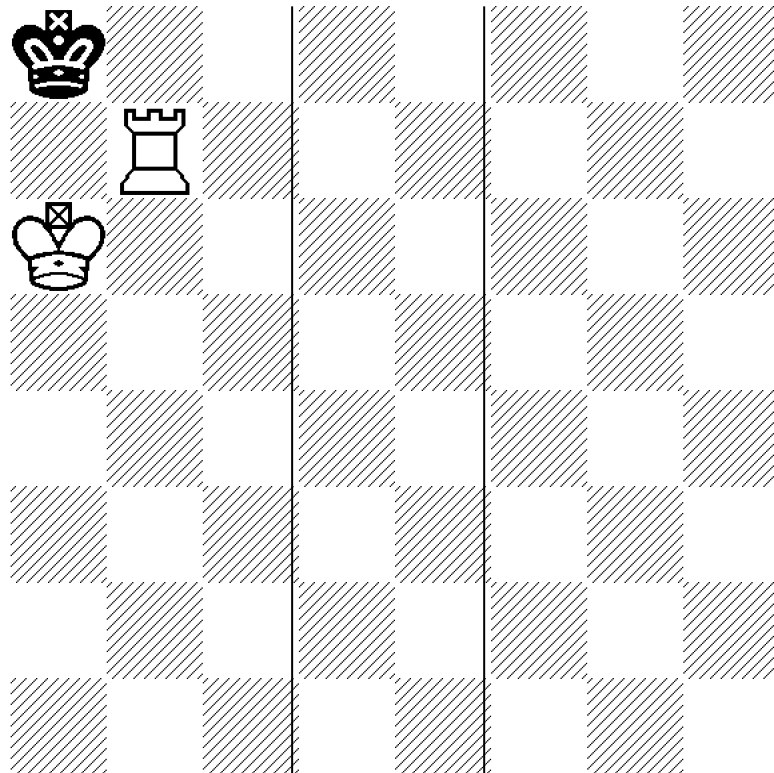


Plate 1. Torre y Quevedo's chess machine (see Chapter 2).

La macchina (1890) dell'ingegnere militare spagnolo Torre y Quevedo (inventore di una micidiale specie di siluro) giocava perfettamente il finale di Re e Torre contro Re.



Cronologia moderna

Negli USA ed in URSS i militari ispirarono e finanziarono le ricerche su teoria dei giochi e sviluppo di giocatori artificiale per molti anni

Il primo programma funzionante è del 1956 (Los Alamos Labs), ma su scacchiera 6x6 (mancavano gli alfieri)

Al MIT, a CMU e a Stanford vennero prodotti i primi programmi completi (1958-64)

Nel 1967 una macchina sovietica (a Mosca) sconfisse 4-0 una macchina americana (a Stanford) giocando per ... corrispondenza!

La prima volta che una macchina partecipò ad un torneo contro umani fu nel 1967 (MacHack di R. Greenblatt, MIT)

Nel 1980 K.Thompson (Bell Labs) costruì Belle, la prima macchina che conseguì in torneo il titolo di Maestro

Nel 1988 Deep Thought (Hsu e Campbell-CMU) sconfisse per la prima volta un Gran Maestro (Larsen) in torneo

Nel 1994 alcuni programmi commerciali (Fritz3, Genius) hanno battuto in alcune partite ufficiali brevi (blitz) il Campione del Mondo (Kasparov)

Nel 1997 Deep Blue di IBM ha battuto di misura il Campione del Mondo (Kasparov) in un torneo-match

Teoria dei giochi

Nel XX secolo nasce un approccio teorico e formale allo studio dei giochi come metafora di situazioni di conflitto

Gli Scacchi sono un gioco a *informazione completa*

- ⇒ due avversari alternano le mosse e conoscono in ogni istante le stesse informazioni sullo “stato del gioco”;
- ⇒ ad ogni turno di gioco le mosse ammesse dalle regole sono ben definite e finite;
- ⇒ la partita termina con la vittoria di uno dei due giocatori, oppure in parità

Il teorema del minimax [Von Neumann e Morgestern] assicura che esiste un risultato univoco, cioè una strategia ottima (la “partita reale”)

Il problema è che esistono circa 10^{120} partite possibili e circa 10^{43} posizioni!

(I processori più veloci oggi eseguono $\sim 10^9$ op/secondo; un anno comprende $\sim 3 \cdot 10^7$ secondi; ammettendo di poter analizzare $\sim 10^9$ posizioni/secondo, occorrono comunque 10^{27} anni per analizzare le posizioni possibili; l’universo ha circa 10^{10} anni)

L'albero di gioco (esplosione combinatoria)

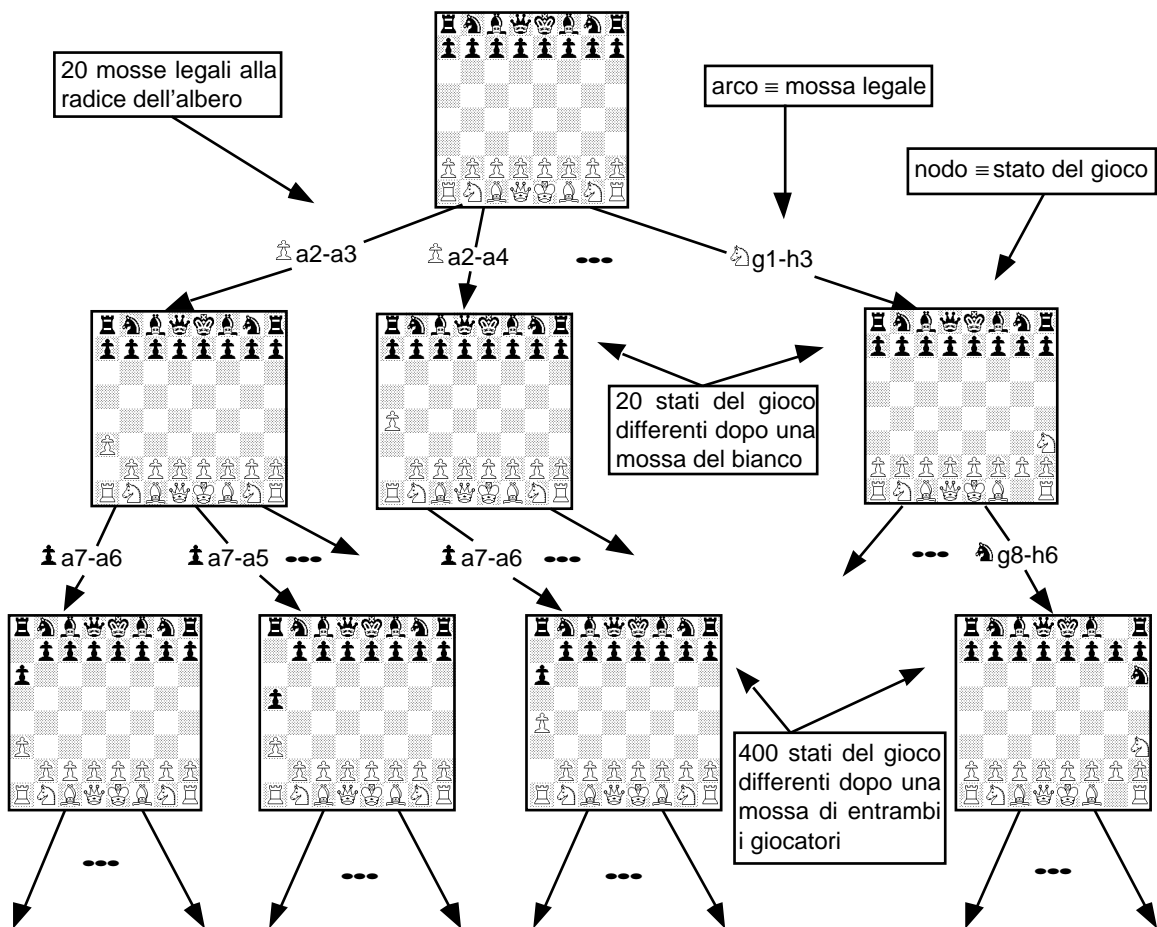
I giochi a informazione completa si possono descrivere mediante un grafo, l'*albero di gioco*, i cui nodi sono posizioni e i cui archi sono mosse legali

radice: posizione iniziale

arco: mossa legale

nodo figlio: posizione raggiungibile con una mossa legale

nodo terminale: posizione terminale (scacco matto, stallo)



Complicazione: l'albero di gioco non è un vero albero, ma un grafo, perché le posizioni si possono ripetere

La ricerca sulle macchine che giocano a Scacchi

All'inizio degli anni '60 un articolo di Newell, Shaw e Simon, tre scienziati americani (Simon avrebbe vinto il Nobel dell'Economia pochi anni dopo) giustificò le ricerche sui giocatori artificiali di Scacchi come segue:

“Gli Scacchi sono il gioco intellettuale per eccellenza.

Senza far uso di strumenti casuali (come i dadi o la roulette), che inquinerebbero la contesa, due intelletti vengono contrapposti in una situazione così complessa che nessuno dei due può sperare di comprenderla completamente, e tuttavia il gioco è sufficientemente analizzabile, di modo che ciascuno dei due può sperare di sconfiggere l'altro.

Il gioco è tanto profondo e sottile che ha permesso la nascita di giocatori professionisti, ed ha sopportato senza esaurirsi oltre 200 anni di partite e di studi analitici intensivi. Tali caratteristiche rendono gli Scacchi un'arena naturale per i tentativi di meccanizzazione.

Se si potesse sviluppare un giocatore artificiale vincente, si potrebbe affermare di aver penetrato il nucleo dell'attività intellettuale umana.”

I centri di ricerca

I centri principali in cui oggi si sviluppa la ricerca sulle macchine che giocano sono:

- IBM Research Center, Yorktown (USA)
 - MIT, Boston (USA)
 - Carnegie Mellon Univ., Pittsburgh (USA)
 - Univ. di Paderborn (Germania)
 - Univ. di Alberta (Canada)
 - Univ. di Maastricht (Olanda)
-
- La ICGA (Intern. Computer Games Association) è una associazione scientifica che pubblica da circa venti anni una rivista accademica trimestrale
 - Esiste una conferenza triennale, *Advances in Computer Chess*, giunta nel 1999 alla 9a edizione
 - Negli anni '80 presso il Politecnico di Milano vennero organizzate tre conferenze su Scacchi e Intelligenza Artificiale
 - Un Torneo di Campionato del mondo per giocatori artificiali si tiene ogni 3 anni

Giocatori artificiali di Scacchi

Perché gli Scacchi sono un gioco interessante da studiare come applicazione di tecniche di Intelligenza Artificiale?

- metafora della guerra e di situazioni di conflitto
- regole semplici ma con spazio di stati esponenziale
- enorme letteratura disponibile
- numerosi professionisti umani contro cui misurarsi
- sistema oggettivo di valutazione della forza di gioco
- mercato commerciale
- scopi pubblicitari (IBM, Intel)
- soprattutto, le tecniche che funzionano per gli Scacchi vengono esportate anche in altri domini (sistemi esperti, dimostratori di teoremi, apprendimento automatico, ecc.)

Intelligenza artificiale?

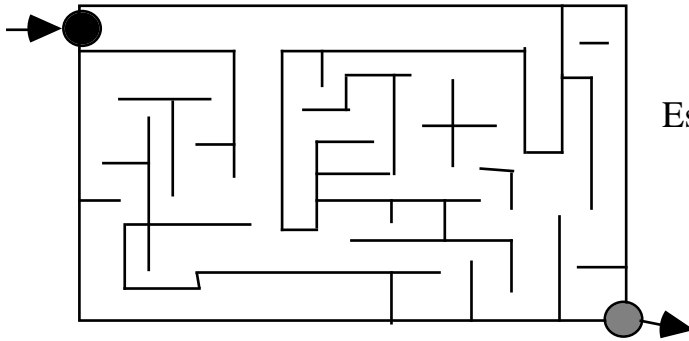
Le tecniche che usano i giocatori artificiali possono essere usate anche in altri domini

La ricerca sulle macchine che giocano giochi di scacchiera è stata finanziata inizialmente in quanto tali giochi (scacchi, dama) sono metafore della guerra

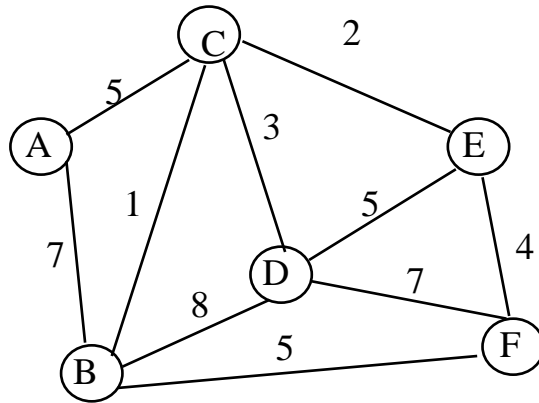
In realtà i risultati scientifici ottenuti sono stati applicati in molti altri campi

- dimostrazione automatica di teoremi
- ricerca operativa (risoluzione di problemi su grafi)
- analisi sintattica
- sistemi esperti
- apprendimento
- robotica

Esplorazione di uno spazio di stati



Esplorazione di un labirinto



Ricerca di cammini su grafi

- TSP
- costo minimo
- ecc.

A -> BcD
B -> Dc
D -> dd

Riconoscimento di
un linguaggio formale

p(a).
r(b).
q(X):- p(X),r(X).

Dimostrazione di teoremi

La misura del progresso

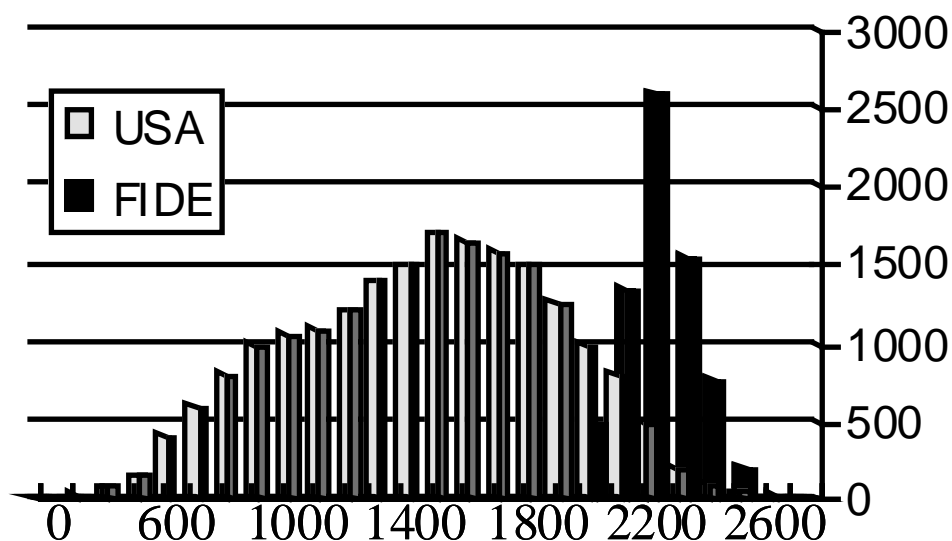
Uno degli aspetti più interessanti della ricerca sulle macchine che giocano a Scacchi è l'esistenza di un sistema di valutazione della forza di gioco (sistema Elo), capace di misurare il progresso dei giocatori artificiali

Il sistema è stato messo a punto da uno statistico, il prof. A.Elo, mediante studi sui principali tornei degli ultimi 150 anni; lo scopo originale è quello di misurare la forza di gioco degli essere umani

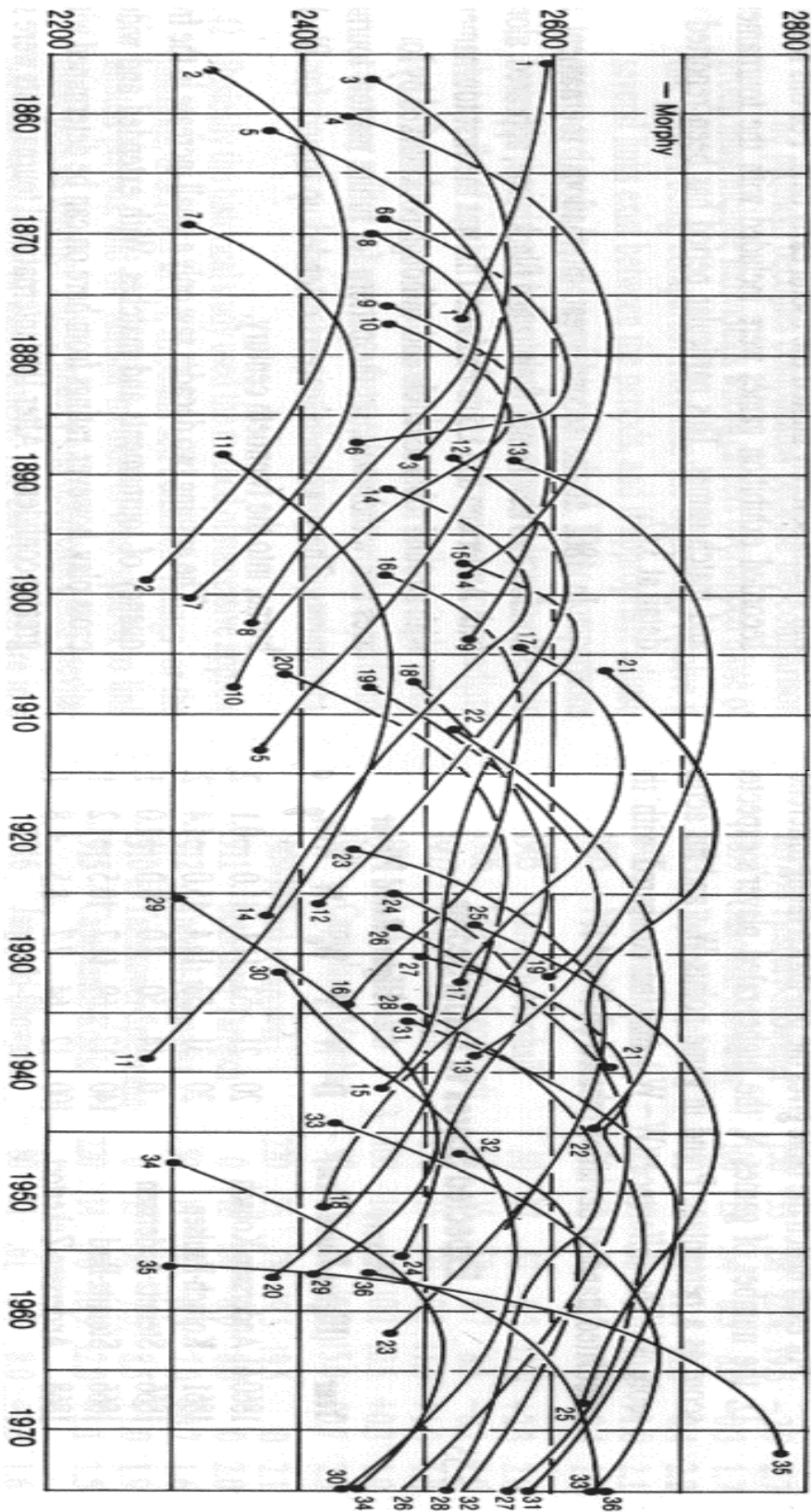
Giocatore principiante	500
Giovane giocatore di club	1000
Media giocatori agonistici	1500
Maestro	2200
Gran Maestro Internazionale	2500
Campione del Mondo	2800

I punti si guadagnano giocando in torneo ufficiale

Distribuzione tipica dei giocatori umani:

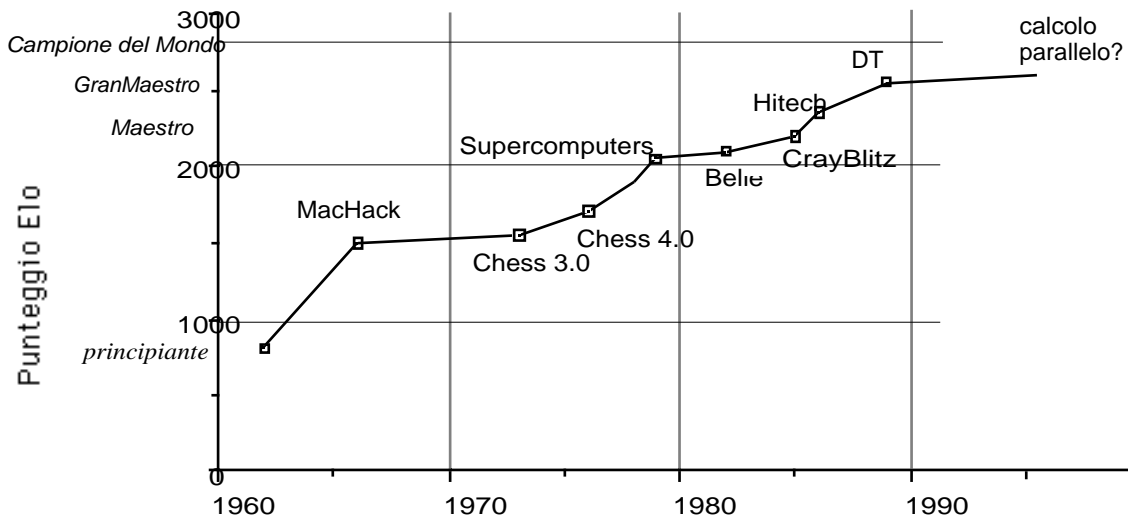


Valutazione storica dei campioni umani



Il sistema Elo applicato alle macchine

Il sistema Elo si può applicare alle macchine che giocano in torneo contro esseri umani



Interfaccia di un giocatore artificiale

MacChess 2.0

PV

a2-a4 Bf8-g7
c7-c5

Time

00:01:21	00:00:10
00:00:43	00:00:03
00:00:05	00:00:06
45	45
00:04:22	00:04:50

Openings Book

Ruy Lopez
Closed Defence
Breyer Variation

end 15

Info

score : -2
depth : 3 25
number mov: 32 32
mov: 32
eval. : 14840
ev./sec.: 4946
MiddleGame

Moves

6. Rf1-e1	b7-b5
7. Ba4-b3	d7-d6
8. c2-c3	0-0
9. h2-h3	Nc6-b8
10. d2-d4	Nb8-d7
11. Nb1-d2	Bc8-b7
12. Bb3-c2	Rf8-e8
13. Nd2-f1	Be7-f8
14. Nf1-g3	g7-g6
15. Bc1-g5	Bf8-g7
16.	

Giocatori artificiali

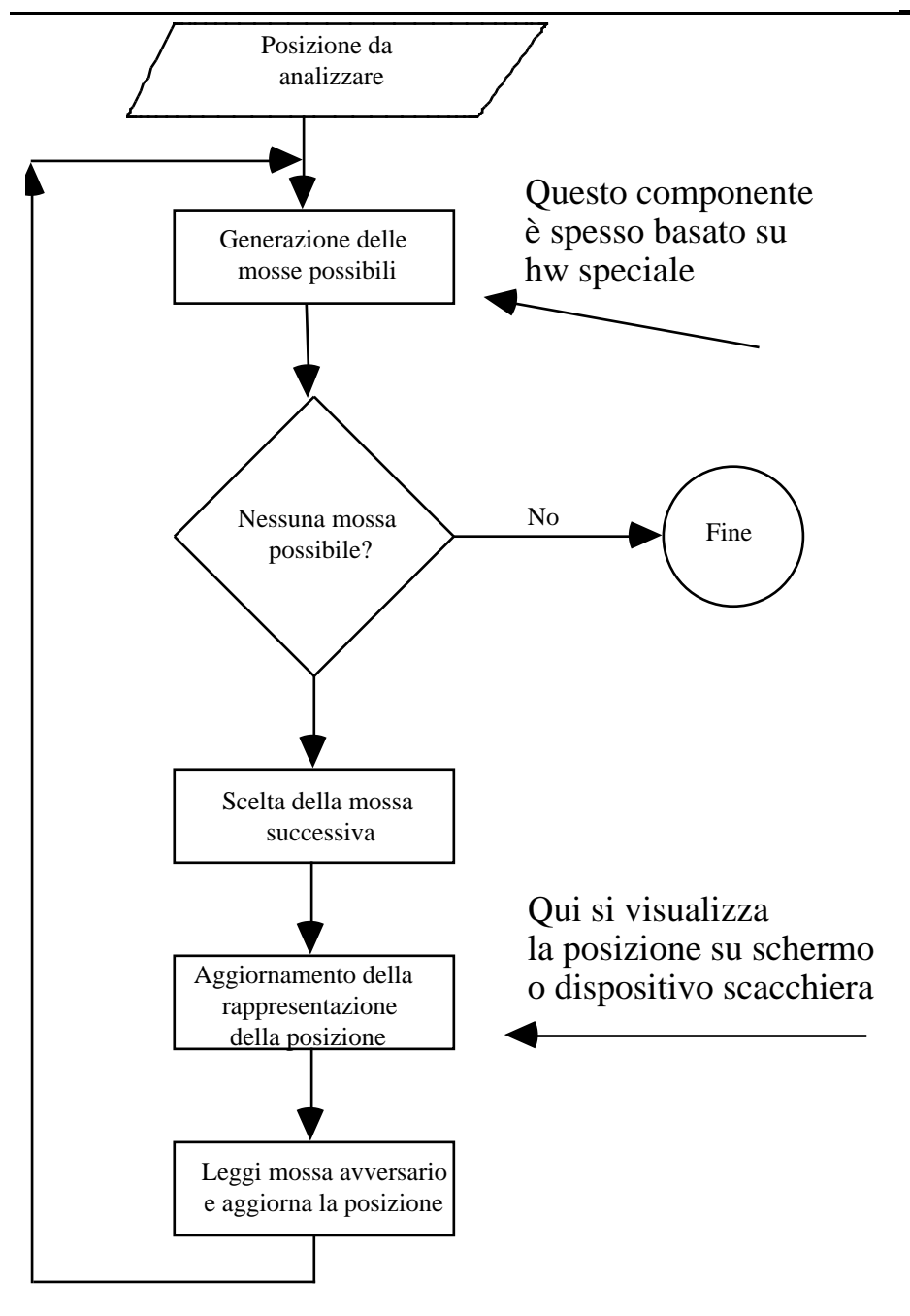
20

Architettura di un giocatore artificiale

I giocatori artificiali esistenti (sia commerciali che di ricerca) possono essere classificati in due categorie:

- quelli che si basano su hardware speciale (es. Deep Blue)
- quelli che usano hardware generico (es. PC IBM)

La struttura software è comunque semplice



Classificazione di Shannon

Nessun programma può esplorare completamente l'albero di gioco degli Scacchi: occorre accontentarsi di visite parziali

Shannon definì due tipi di strategie per gestire visite parziali

Strategia di tipo A: esplorazione per "*forza bruta*"

Per ogni posizione si valutano tutte le mosse possibili, fino ad un certo livello dell'albero di gioco (effetto "orizzonte")

Strategia di tipo B: esplorazione *con valutazione euristica*

Per ogni posizione si valutano solo le mosse "interessanti", fino ad un certo livello (minore di quello di tipo A)

Per i primi venti anni (1955-75) la ricerca si è concentrata su programmi di tipo B, tentando di definire euristiche efficienti di selezione delle mosse interessanti e metodi di pianificazione

Purtroppo, non fu mai scoperta una combinazione di euristiche sufficiente per giocare in modo "intelligente"

“Al 99%, gli Scacchi sono tattica”

I grandi successi dei giocatori artificiali si sono verificati a partire dagli anni '80, quando i progressi delle tecnologie VLSI di costruzione dei microprocessori hanno permesso il progetto di programmi di tipo A sufficientemente efficienti

Macchina	Posizioni/secondo	HW
MChess	10.000	Pentium 90
Zarkov	15.000	HP735
Cray Blitz	750.000	Cray C90
Deep Thought 2	4.000.000	IBM/6000
Deep Blue	200.000.000	IBM SP/32

Oggi su PC si arriva ad esplorare alberi profondi 12/14 semimosse in 3 minuti, il tempo medio per mossa di torneo;

Si calcola che ogni semimossa in più fa guadagnare circa 50/100 punti Elo (esperimento di Thompson)

Incremento di forza in funzione della profondità di analisi

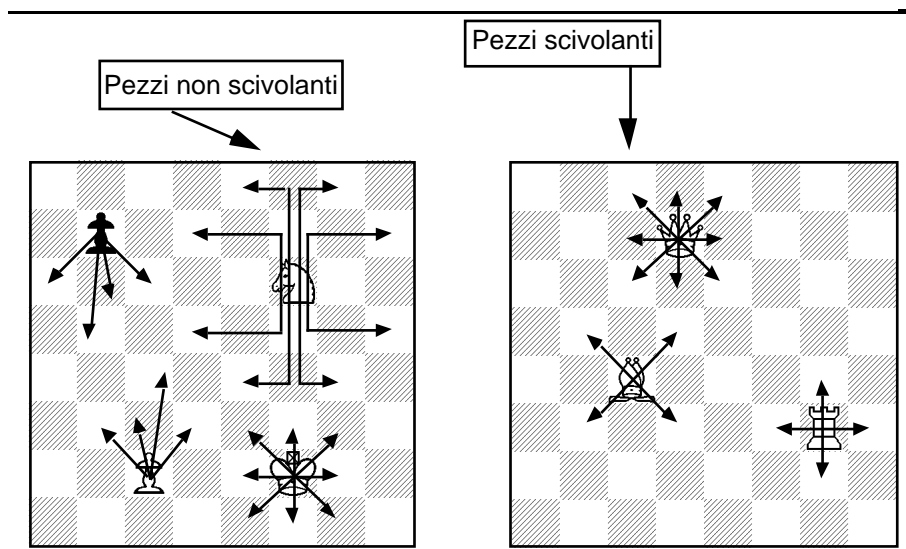
	P4	P5	P6	P7	P8	P9	Elo
P4	-	5	0.5	0	0	0	1235
P5	15	-	3.5	3	0.5	0	1570
P6	19.5	16.5	-	4	1.5	1.5	1826
P7	20	17	16	-	5	4	2031
P8	20	19.5	18.5	15	-	5.5	2208
P9	20	20	18.5	16	14.5	-	2328

Generazione delle mosse

L'operazione cardine nella creazione dell'albero di gioco è la generazione di tutte le mosse legali in una data posizione

Un metodo di calcolo molto efficiente si basa su tabelle precalcolate di *mappe dei movimenti*: riportano tutte le mosse possibili per ogni pezzo a partire da una casa qualsiasi e senza tenere conto della presenza di altri pezzi

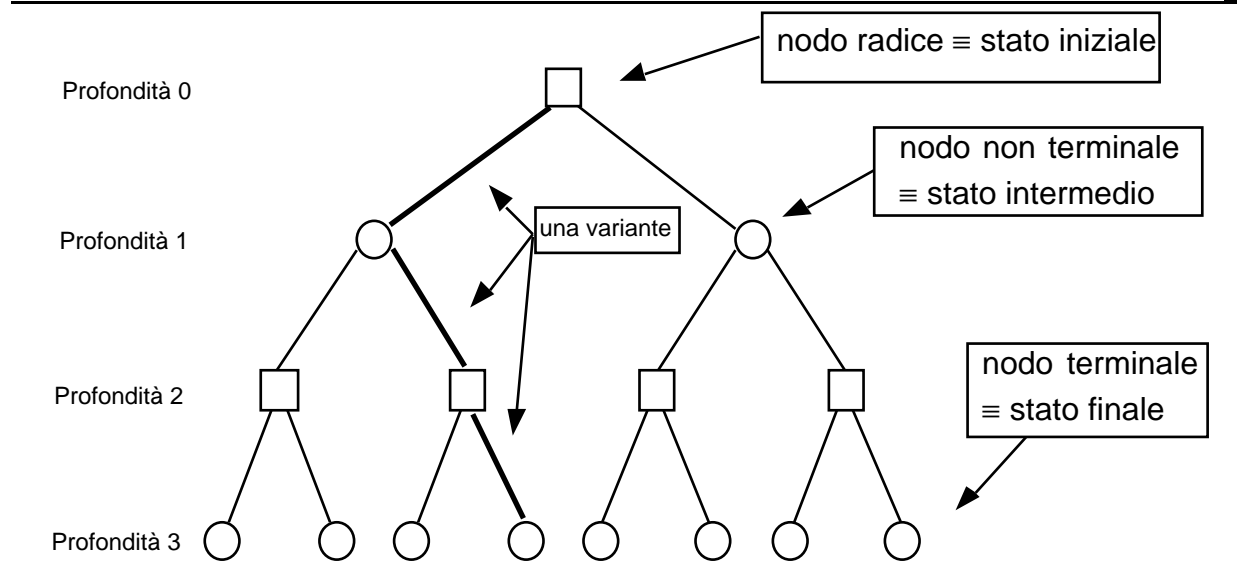
Il calcolo di questi dati utilizza dei *vettori di movimento*: sono liste che per ogni pezzo definiscono la relazione matematica fissa esistente fra le coordinate di partenza e le possibili coordinate di arrivo



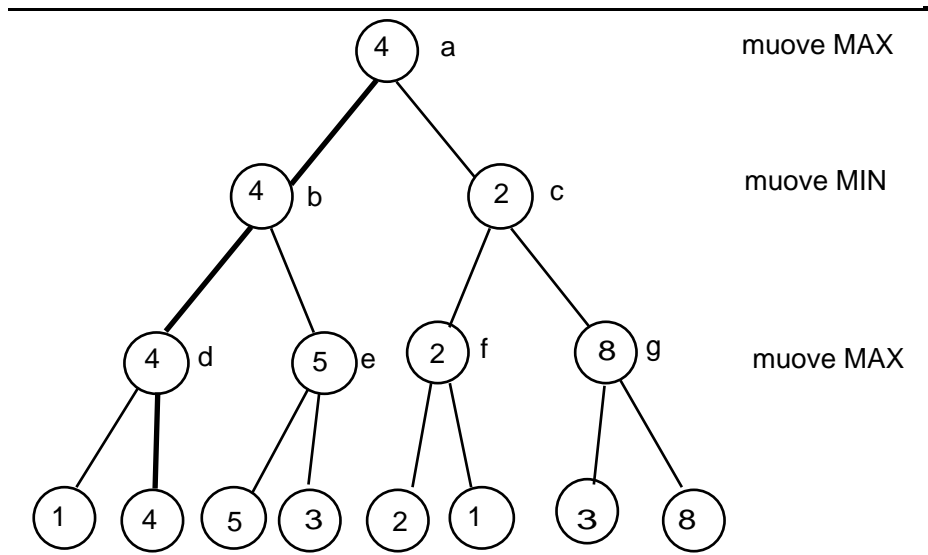
Pezzo	Vettore di movimento (k è la casa di partenza)
	k+7, k+8, k+9, k+16
	k+6, k+10, k+15, k+17, k-6, k-10, k-15, k-17
	k+7, k+9, k-7, k-9 e così via
	k+1, k+8, k-1, k-8 e così via
	k+1, k+7, k+8, k+9, k-1, k-7, k-8, k-9 e così via
	k+1, k+7, k+8, k+9, k-1, k-7, k-8, k-9
	k-7, k-8, k-9, k-16

Valutazione di un albero di gioco

Il generatore di mosse permette di costruire l'albero di gioco, che deve essere valutato a partire dalle foglie



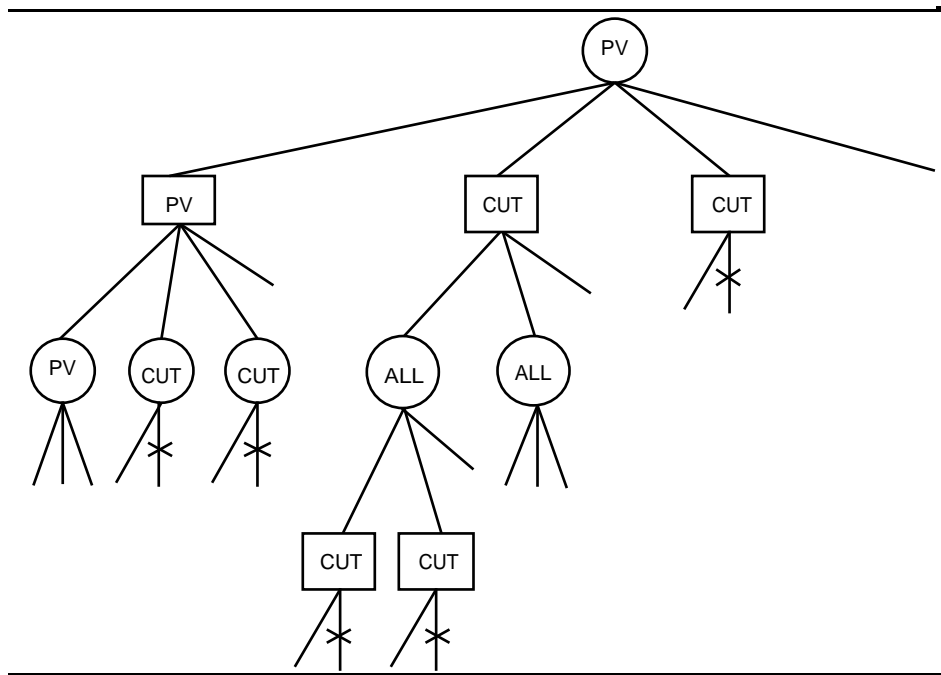
L'algoritmo *minimax* è l'esempio più importante di strategia di controllo per la classe di giochi finiti a due giocatori a informazione completa



Algoritmi di ricerca più efficienti

Data una posizione p e l'albero di gioco a partire da p , lo scopo di un algoritmo di ricerca è quello di determinare il valore minimax del nodo p ; questo valore sarà una approssimazione di quello reale nel caso che l'albero visitato sia *parziale*, cioè i valori associati ai suoi nodi foglie non siano esatti, ma frutto di una stima

L'algoritmo $\alpha\beta$ produce lo stesso risultato della ricerca minimax, ma ad un minor costo computazionale. L'algoritmo effettua dei *tagli* sull'albero di gioco, ovvero determina sottoalberi che non contengono il valore ottimo per la radice e quindi ne evita la visita



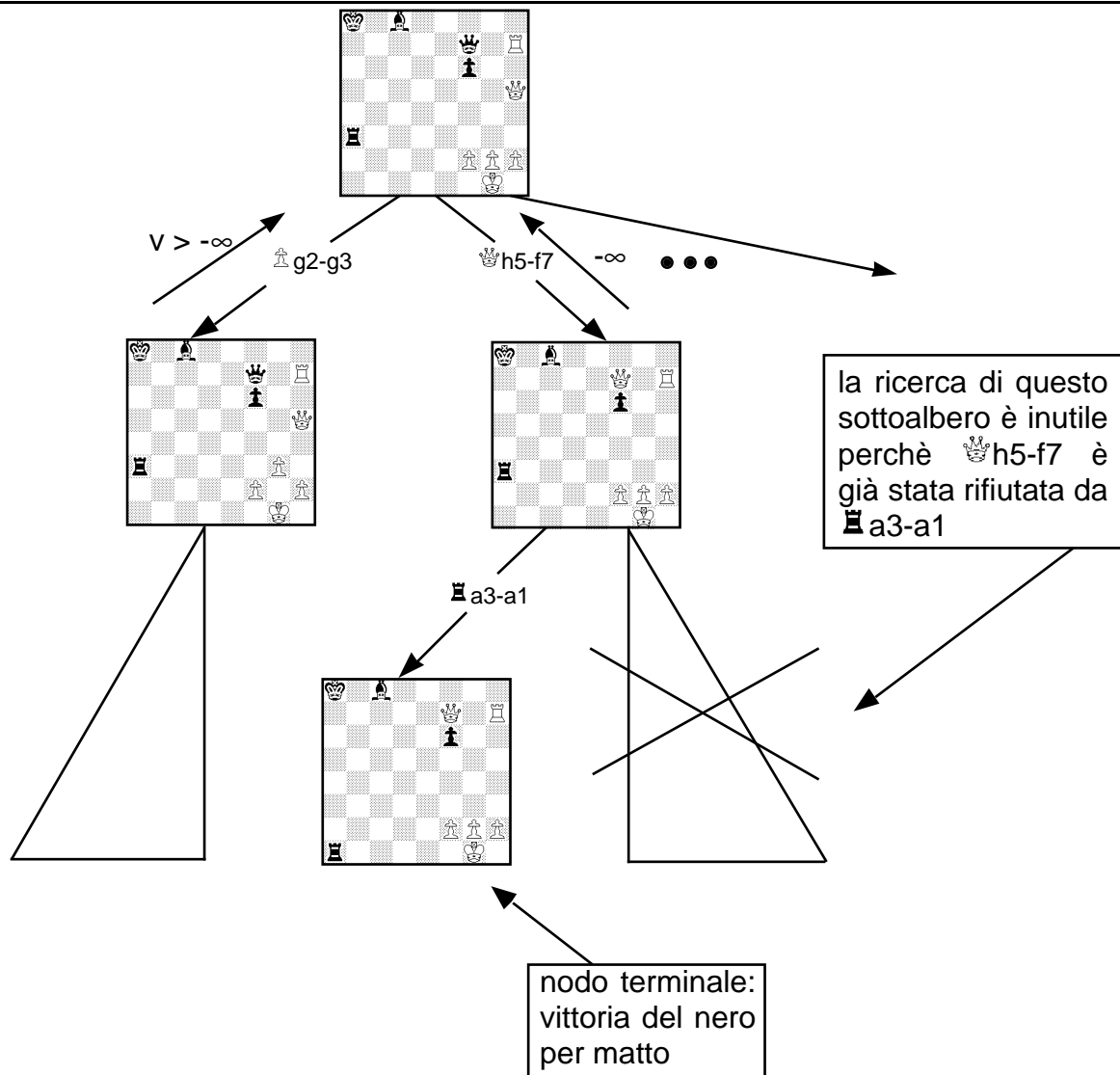
Nel caso pessimo alphabeta si comporta come minimax e valuta pertanto w^d nodi terminali

Si dimostra che per alberi con attribuzione casuale dei valori ai nodi terminali, nel tempo che minimax completa una ricerca a profondità d , alphabeta esplora in media un albero di profondità $4/3 d$ [Pea82]

Alfabeta: esempio

La prima mossa valutata dal bianco è ♖g2-g3 il cui valore minimax è $v > -\infty$

La successiva mossa valutata è ♔h5-f7; in risposta a tale mossa è analizzata la mossa ♜a3-a1 la quale origina immediatamente il matto per il nero: indipendentemente dalle altre mosse di risposta del nero il valore minimax finale per la mossa ♔h5-f7 equivale ad una sconfitta ($= -\infty$) e quindi questa non è una scelta conveniente per il bianco.



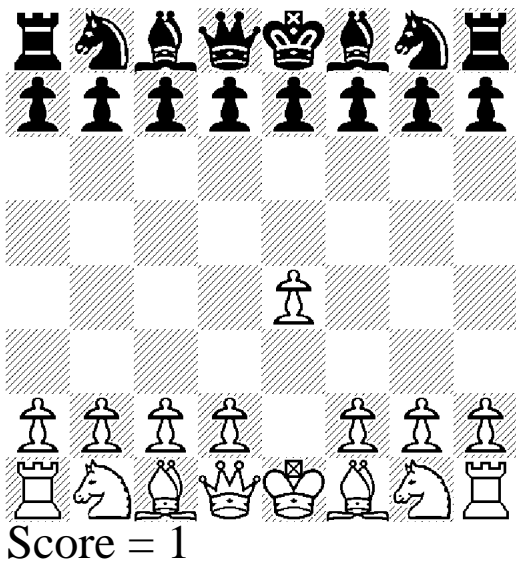
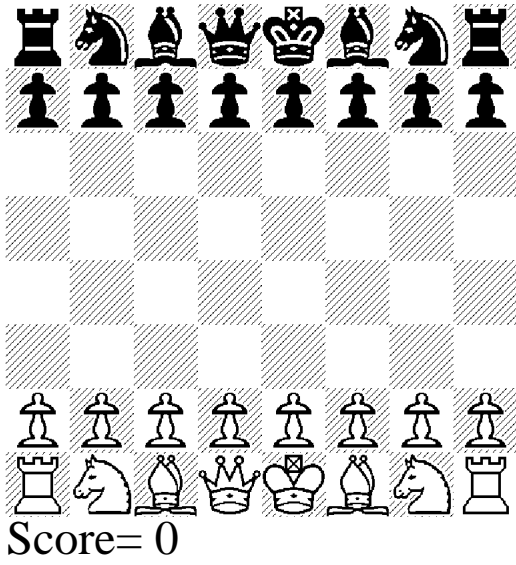
Funzione di valutazione

Shannon propose di usare una funzione $S(\text{Score})$ per valutare una posizione Pos :

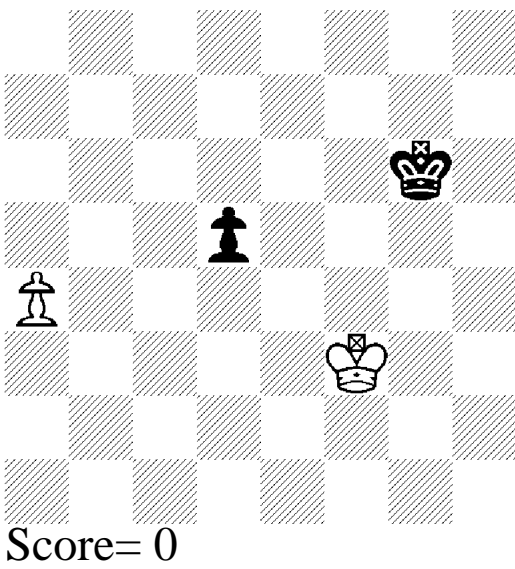
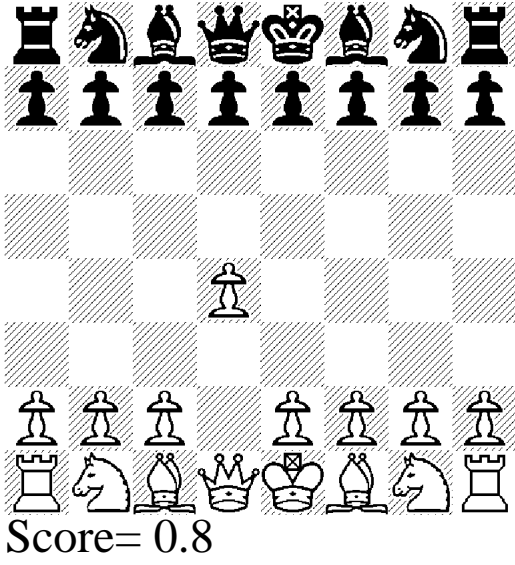
- materiale ($P=1, D=9, R=5, A=C=3, R=200$)
- mobilità: somma del n. di mosse legali per ciascun pezzo
- penalità per ogni pedone doppiato, arretrato o isolato

$$S(Pos) = 200(k-K) + 9(d-D) + 5(r-R) + 3(a-A+c-C) + (p-P) + \\ + 0.1(\text{mob-MOB}) + \\ + 0.5(\text{dop-DOP} + \text{arr-ARR} + \text{iso-ISO})$$

Funzione di valutazione: esempi



Funzione di valutazione: esempi







Una funzione di valutazione: GNU Chess

Uno dei più diffusi programmi di gioco è GNUchess

Selezione della mossa:

- aspiration search fail-soft
- approfondimento iterativo
- ricerca quiescente
- tabella delle trasposizioni
- euristica dei killer
- libro di aperture

Funzione di valutazione:

 = 100	 = 330
 = 330	 = 520
 = 980	 = 10000

- materiale
- spazio e mobilità
- sicurezza del Re
- controllo del centro
- struttura pedonale
- sistemazione dei pezzi
- possibilità di attacco
- relazione tra i pezzi

Un esperimento per valutare l'importanza dei vari fattori di conoscenza [Schaeffer]:

versione 1: solo materiale	1110 p. Elo
versione 2: vers. 1 + mobilità e spazio	1420 p. Elo
versione 3: vers. 2 + controllo del centro	1530 p. Elo
versione 4: vers. 3 + struttura pedonale	1600 p. Elo
versione 5: vers. 4 + valutazione mosse	1630 p. Elo
versione 6: vers. 5 + sicurezza del Re	1750 p. Elo
versione 7: vers. 6 + case deboli e P passati	1760 p. Elo
versione 8: vers. 7 + pianificazione	1780 p. Elo

Uso di database e apprendimento

- Database di aperture
- Database di finali
- Database di posizioni e partite

Problemi:

- Gestione di grandi quantità di informazioni
- Reperimento in tempo reale di una posizione
- Uso della conoscenza implicita nelle partite magistrali

Alcune tecniche utilizzabili:

- Apprendimento automatico
- Data mining
- Algoritmi genetici

Influenza dell'hardware sulla forza di gioco

L.Kaufmann ha usato il programma GNUChess testandone le prestazioni su elaboratori di diversa potenza, allo scopo di valutare l'incidenza delle prestazioni hw sulla forza di gioco

Kaufmann prese come termine di confronto il vecchio IBM AT nella versione con processore Intel da 8 Mhz, capace di eseguire un milione di istruzioni per secondo (MIPS), dove le istruzioni erano quelle risultanti dalla compilazione del programma GNUChess, che è scritto in linguaggio C

Allora un IBM XT con processore 8088 a 4.77 Mhz vale 0.25 MIPS, un AT con processore 286 (tipo "0 wait state") vale 2 MIPS, un AT 386 (33 Mhz, con memoria Cache) vale 7 MIPS, un AT 486 a 25 Mhz vale 11 MIPS, ecc.

I processori Motorola sono meno ottimizzati rispetto al C, ma hanno un assembler comunque molto efficiente

Alcuni dati per macchine con processore Motorola: 68020 vale 3.2, il 68030 vale circa 11, mentre un 68040 vale quasi 20 MIPS

Per quel che riguarda il "vecchio" processore 6502, cuore del Commodore 64 e di tante scacchiere elettroniche di tipo economico, nella versione da 6 Mhz vale 1.7 MIPS (nel C64 la frequenza è di 1 Mhz, il che vale 0.25 MIPS)

Influenza dell'hardware sulla forza di gioco

Le valutazioni di Kaufmann hanno anche un valore predittivo. La tabella mostra la variazione presunta in forza Elo in termini della sola velocità hw calcolata in MIPS

MIPS	0.25	0.5	1	1.5	2	3	4	6
Δ Elo	-180	-87	0	47	80	124	154	195

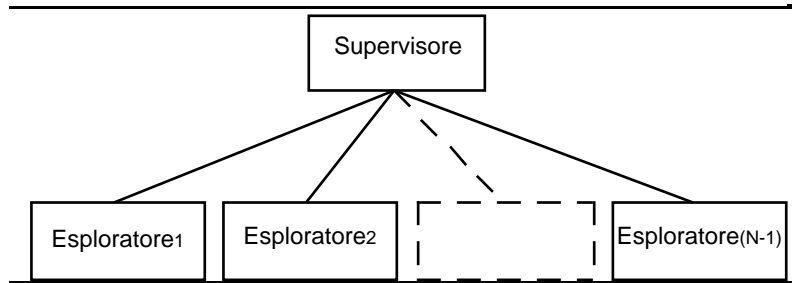
MIPS	8	12	16	24	32	48	64
Δ Elo	223	261	287	323	347	379	402

Quindi lo stesso programma che gira su una macchina con processore 80486 a 33 Mhz (15 MIPS) è 60 volte più veloce, e vale all'incirca 430 punti in più, che su un XT

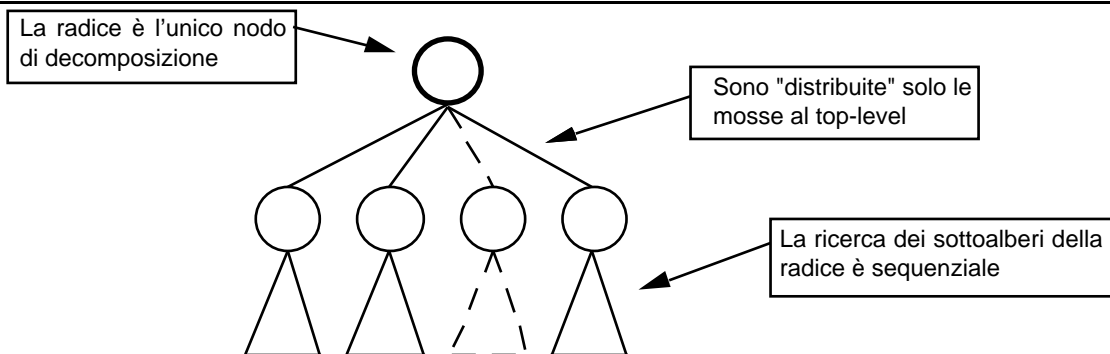
Per il 2000 si prevedono processori da 1000 MIPS, per cui si prevede un incremento Elo pari a 580 punti Elo

Esplorazione parallela

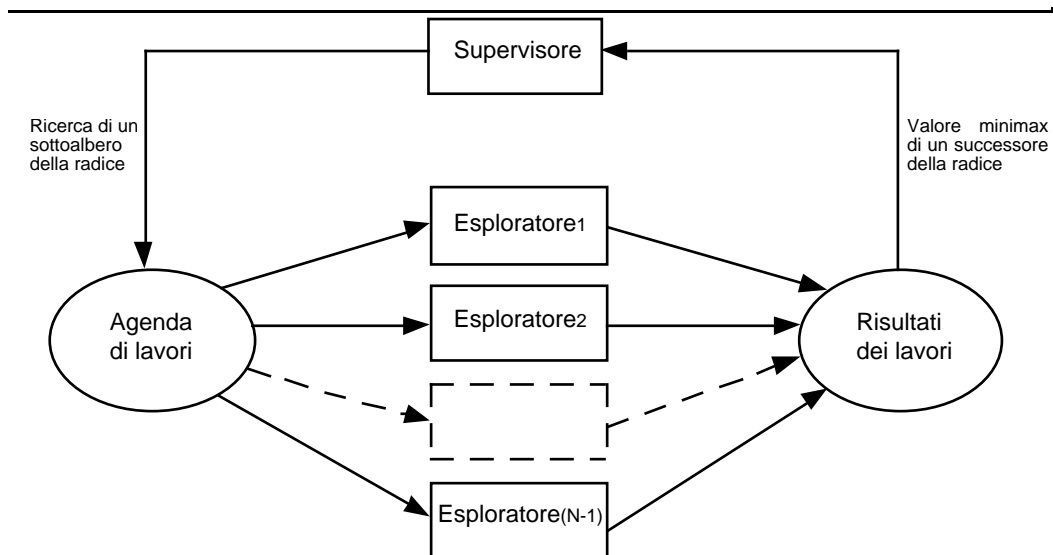
L'esplorazione parallela di alberi di gioco è una buona idea?



Architettura distribuita per algoritmi paralleli di ricerca



Decomposizione dell'albero di gioco



Struttura logica della comunicazione

Influenza di memoria e potenza di calcolo

[Kusmaul 94] contiene alcuni dati sull'uso di grandi quantità di memoria e processori, ottenuti col programma *tech (versione C di Hitech) su CM5

Influenza della tabella delle trasposizioni (un solo processore, RAM sufficiente per tabella con 2^{22} posizioni)

Table size	Posiz visitate	Tempo (sec)	Elo stimato
0	161.600	23.337	2056
2^{16}	94.400	13.506	2104
2^{17}	85.750	12.670	2109
2^{18}	76.500	10.925	2122
2^{19}	65.600	9.605	2133
2^{20}	55.900	8.040	2149
2^{21}	48.250	7.138	2159
2^{22}	42.630	5.799	2177
2^{23}	40.800	6.120	2173
2^{24}	40.800	6.364	2169

Forza stimata in funzione del numero di processori (CM5, 32 Mb per proc., tabella hash 2^{21} pos per processore)

Processori	Tempo (sec)	Elo stimato
1	8.936	2139
2	5.376	2183
4	3.152	2230
8	1.932	2272
16	1.240	2311
32	844	2344
64	573	2378
128	444	2400
256	378	2414
512	319	2429

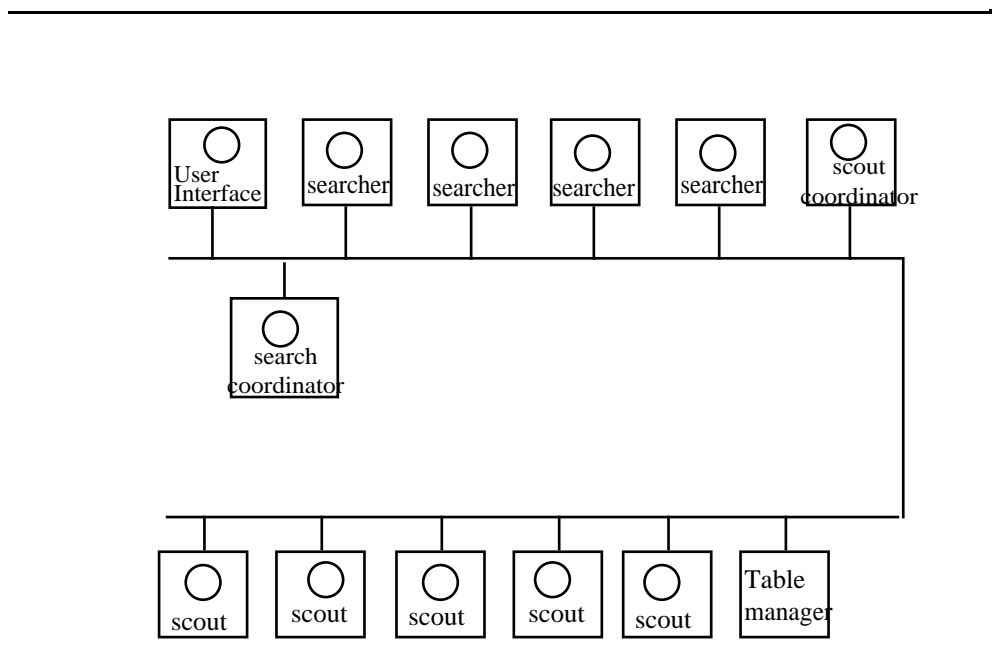
Distribuzione e coordinazione della conoscenza

Parallelizzare la ricerca alfabetica è difficile; specie su rete non si ottengono guadagni consistenti oltre i 10 processori

Idea: invece di parallelizzare la ricerca, provare a coordinare la distribuzione di conoscenza tra diversi agenti

Esperimenti:

- Schaeffer (1987)
- Althofer (1991)
- Ciancarini et al. (1993)



- Assegnazione di porzioni di conoscenza diversa a più unità di elaborazione: le unità con meno conoscenza esplorano l'albero più in profondità
- Ogni unità propone una mossa: occorre un metodo di valutazione e scelta di una delle mosse proposte

Come distribuire la conoscenza?

Quale criterio di selezione applicare? come ottimizzare?

Galleria di campioni artificiali

Macchine sviluppate a scopi di ricerca

- Deep Blue (ex Deep Thought) [Hsu e Campbell]
- Hitech [Berliner]
- Crafty, Cray Blitz [Hyatt]
- Belle [Thompson]

Programmi commerciali (tutti su PC)

Valutazione di alcuni programmi commerciali (Feb/1998)

(a cura della Federazione Svedese: www.nsc.liu.se/~bosj/SSDF)

1.Fritz 5.0 Pentium 200 MMX	2589	*(autotuning)
2.Nimzo 98 Pentium 200 MMX	2534	
3.Hiarcs 6.0 Pentium 200 MMX	2533	
4.Rebel 9.0 Pentium 200 MMX	2528	
5.MChessPro 7.1 Pentium 200 MMX	2523	

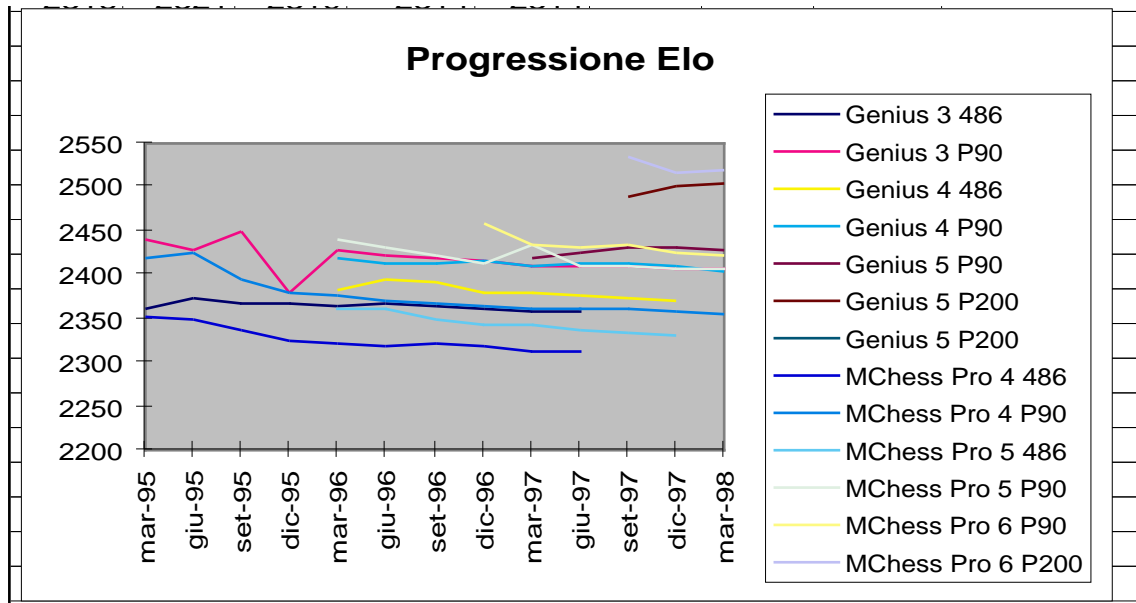
Esaminiamo le differenze di Elo di uno stesso programma:

1.Fritz 5.0 P200 MMX	2589
2.Fritz 3.0 P90	2347
3.Fritz 4.0 P90	2335

1.Hiarcs 6.0 P200 MMX	2533
2.Hiarcs 6.0 P90	2450
3.Hiarcs 4.0 P90	2394

1.MChessPro 7.1 P200 MMX	2523
2.MChessPro 6 P200 MMX	2519
3.MChessPro 6 P90	2422
4.MChessPro 5 P90	2407
5.MChessPro 4 P90	2356

Analisi diacronica dell'ELO di alcuni programmi



	mar-95	giu-95	set-95	dic-95	mar-96	giu-96	set-96	dic-96	mar-97	giu-97	set-97	dic-97	mar-98	
Genius 3 486	2362	2372	2367	2366	2363	2366	2364	2360	2358	2359				
Genius 3 P90	2440	2429	2451	2381	2429	2421	2420	2416	2410	2411	2410	2408	2407	
Genius 4 486						2384	2396	2392	2380	2379	2377	2374	2371	
Genius 4 P90						2418	2412	2413	2416	2411	2412	2412	2410	2405
Genius 5 P90										2419	2424	2430	2431	2429
Genius 5 P200												2489	2501	2504
MChess4 486	2352	2350	2336	2324	2322	2318	2321	2319	2314	2314				
MChess4 P90	2418	2424	2396	2381	2376	2369	2367	2363	2360	2360	2360	2358	2356	
MChess5 486					2362	2360	2350	2343	2343	2336	2333	2331		
MChess5 P90					2440	2430	2421	2414	2435	2411	2410	2408	2407	
MChess6 P90									2459	2435	2432	2433	2422	
MChess6 P200												2536	2517	2519

Alcune statistiche

Diamo alcune informazioni su alcuni programmi recenti

Nome	Linguaggio	Pos/se	HW	Età	Pos.note
Cray Blitz	FORTRAN/C/Ass	750.000	Cray C90	15 anni	300.000
Deep Thought 2	C/microass	4.000.000	IBM/6000	7 anni	DB finali
MChess	C+ass	9.000	Pentium 90	6 anni	300.000
*Socrates	C	1.000.000	CM5 / 512	1 anno	13.000
Ananse	TurboPascal	6.000	486/66	8 anni	35.000
Alpha1	C	7.000	Sparc10	6 mesi	50.000
Amy	C	10.000	Sparc10	1 anno	100.000
Arthur	C	20.000	Sparc20	2 anni	10.000
Bobby 2	C	5.000	Pentium 90	7 anni	40.000
Breakthrough	C	15.000	Sparc 20	5 anni	140.000
Cheiron	C	8.000	Sparc 10	6 mesi	11.000
Diogenes	Pascal/Ass.	30.000	Pentium 90	1 anno	30.000
ISIChess	C++/Ass	12.000	486/66	6 mesi	30.000
Zarkov	C	15.000	HP735	8 anni	50.000
ZZZZ	C	4.000	Sparc 20	3 anni	200.000
GNUTeam	C/Linda	10.000	10 Sparc1	2 anni	10.000

Giocare su Internet

È possibile giocare su Internet sia contro avversari umani che contro programmi;

Molti tra programmi più forti si avvalgono di questo strumento per la sperimentazione

I server principali sono:

chessclub.com

fics.chess.org

Prospettive di ricerca

- Sviluppo di nuove architetture hw
- Uso di architetture ad alte prestazioni (Es.: DeepBlue girava su SP/32 con 512 processori speciali)
- Sviluppo di nuovi algoritmi di valutazione e coordinazione paralleli/distribuiti
- Integrazione di conoscenza: basi di dati di aperture, di finali, di piani di gioco
- Ottimizzazione della funzione di valutazione (con algoritmi genetici/reti neurali, su database di partite magistrali)

Comunque, gli Scacchi “ortodossi” sono un problema "quasi risolto": anche se non sarà semplice battere il campione del mondo, ormai solo pochi giocatori umani possono competere con i programmi attuali

Alcuni giochi degni di investigazione nel prossimo millennio:

- Scacchi “Fischer Random”
- Go
- Varianti eterodosse degli Scacchi (es. Kriegspiel)

Aumentare la conoscenza

- Libro delle aperture:
creare il libro delle aperture “automaticamente” da un database di partite
- Euristiche di preordinamento delle mosse:
sviluppare euristiche di ordinamento basate su analisi di aperture ricavate dal database di partite
- Funzione di valutazione:
calibrare i pesi delle euristiche di valutazione parziale attraverso l’analisi delle mosse giocate da maestri (apprendimento)
- Database per alcuni tipi di finale:
ricavati con analisi esaustive, non da database di partite

Conclusioni

Gli Scacchi sono un problema "quasi" risolto sul piano ingegneristico; Poe aveva ragione

Nota: per giochi più semplici (come la Dama) il campione del Mondo è oggi una macchina (Chinook)

Per incrementare la forza di gioco, i progressi hardware sono stati storicamente più importanti delle innovazioni software

Tuttavia

- continua a mancare una “teoria generale” del gioco artificiale, che spieghi successi e fallimenti e soprattutto la miriade di risultati sperimentali disponibili

- Problema aperto:

date le sole regole di un gioco, sviluppare strategie ottimali

Riflessioni finali:

Lo sviluppo di nuove tecniche software (programmazione parallela/distribuita, analisi di grandi database di partite magistrali, apprendimento, integrazione di metodi algoritmici ed euristici) sarà nel prossimo futuro tanto importante quanto i progressi delle tecnologie dell'hardware

I programmi commerciali sono sorprendentemente forti, e non si sa nulla sulla loro struttura: la competizione commerciale “spinge” la tecnologia più del libero scambio di idee?

Bibliografia essenziale

Libri

- Ciancarini P. *Giocatori Artificiali*, Mursia 1992
- Frey P. (ed.), *Chess Skill in Man and Machine*, Springer, 1978 e 1983
- Levy D. (ed.), *Computer Chess Compendium*, Batsford, 1988
- Levy D., Newborn M., *How Computers Play Chess*, 1991.

Articoli

- Akl S., Barnard D., Doran R., "Design, Analysis, and Implementation of a Parallel Tree Search Algorithm", *IEEE Trans. Pattern Analysis and Machine Intelligence*, 4:2, 1982, 192-203.
- Anantharaman T., Campbell M, Hsu F., "Singular Extensions: Adding Selectivity to Brute Force Searching", *Artificial Intelligence*, 43, 1990, 99-110.
- Campbell MS., Marsland TA., "A Comparison of Minimax Tree Search Algorithms", *Artificial Intelligence*, 20:4, 1983, 347-367.
- Ebeling C., *All the Right Moves: A VLSI Architecture for Chess*, PhD Thesis, MIT Press 1987.
- Finkel RA., Fishburn J., "Parallelism in Alpha-Beta Search", *Artificial Intelligence*, 19, 1982, 89-106.
- Knuth DE., Moore RE., "An Analysis of Alpha-beta Pruning", *Artificial Intelligence*, 6, 1975, 293-326.
- Marsland TA., Campbell M., "Parallel Search of Strongly Ordered Game Trees", *ACM Computing Surveys*, 14:4, 1982, 533-551.
- Marsland TA., Popowich F., "Parallel Game Tree Search", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 4:7, 1985, 442.
- Pitrat J., A Chess Combinaton Program which Uses Plans, *Artificial Intelligence*, 8, 1977, 275-321.
- Schaeffer J., Experiments in Search and Knowledge, PhD Thesis, University of Alberta, 1986.