

# Sztuczna Inteligencja

## 4.2 Gry i programy oparte na szukaniu

Włodzisław Duch

Katedra Informatyki Stosowanej UMK

Google: Wlodzislaw Duch

# Co będzie

- Algorytmy szukania w grach.
- Gry planszowe: warcaby, Othello, szachy, go.
- Szukanie i ludzkie myślenie.
- Paradoksy kognitywne.
- Reprezentacja wiedzy ...

# Gry

Gry były dobrym polem do testów dla metod szukania w AI. Prosta reprezentacja: stany, operatory, strategie szukania, prosty cel: znaleźć najlepsze posunięcie.

Problemy:

- niepewność związana z ruchem przeciwnika;
- duża przestrzeń szukania.

Przez długi okres było mało pieniędzy na badania i rozwój programów do gier, z wyjątkiem szachów. Obecnie gry komputerowe są tu siłą napędową. Symulatory do gry w szachy zyskały popularność.

Wyzwaniem AI stało się osiągnięcie mistrzostwa w szachach. Symulatory do gry w szachy zyskały popularność. Alex Bernstein stworzył pierwszy program szachowy już w 1958 roku - był to słaby program.

International Computer Games Association (ICGA) organizuje olimpiady gier komputerowych, np. Abalone, do Surakarta; szachy, go, othello i shogi.

Kompromis pamięć/złożoność ocen.

# Strategia Minimaxu

Teoria gier: von Neumann, Morgenstern 1944

Oponenci w grze: Min i Max – zaczynający.

1. Utwórz drzewo dla gry do maksymalnej głębokości, na jaką cię stać.
2. Oceń wartości f. heurystycznej poczynając od liści.
3. Cofnij się o jeden poziom i dokonaj ocen znajdujących się tam węzłów.
4. Po osiągnięciu korzenia wybierz decyzję maksymalizującą zyski.

Jest to decyzja min-maks: dla skończonych drzew kompletna, dla racjonalnych oponentów najlepsza.

Utwórz tak duże drzewo, na ile starczy czasu.

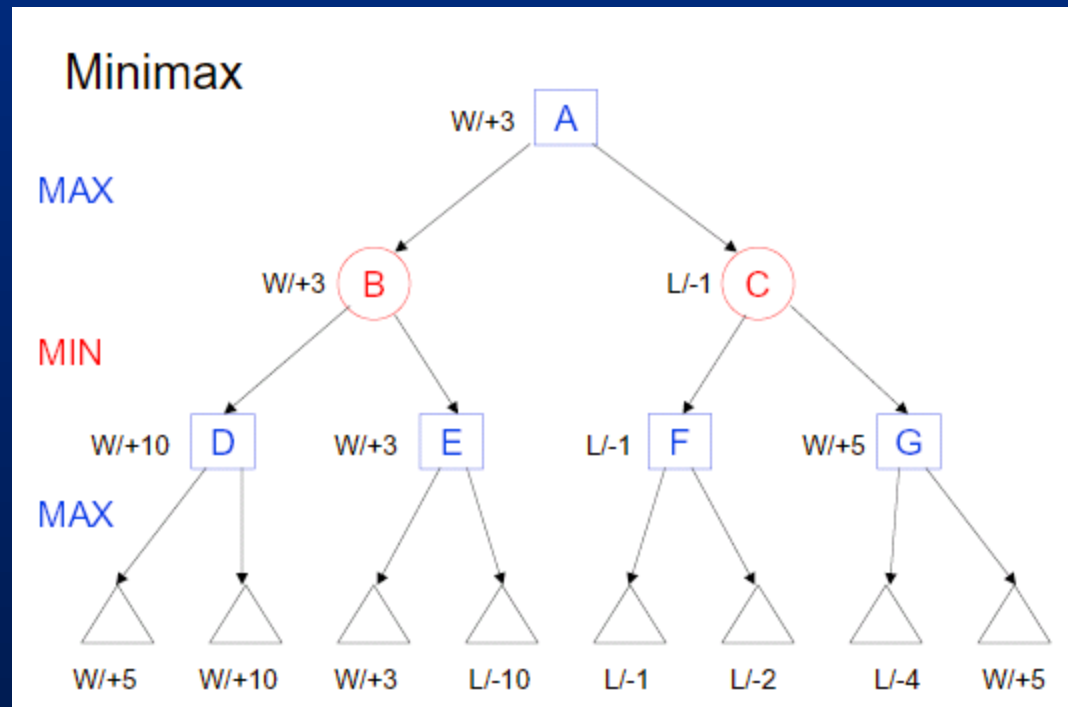
Złożoność  $t \sim O(b^m)$ , pamięć  $O(bm)$  przy szukaniu w głąb.

# Przykład mini-maxu

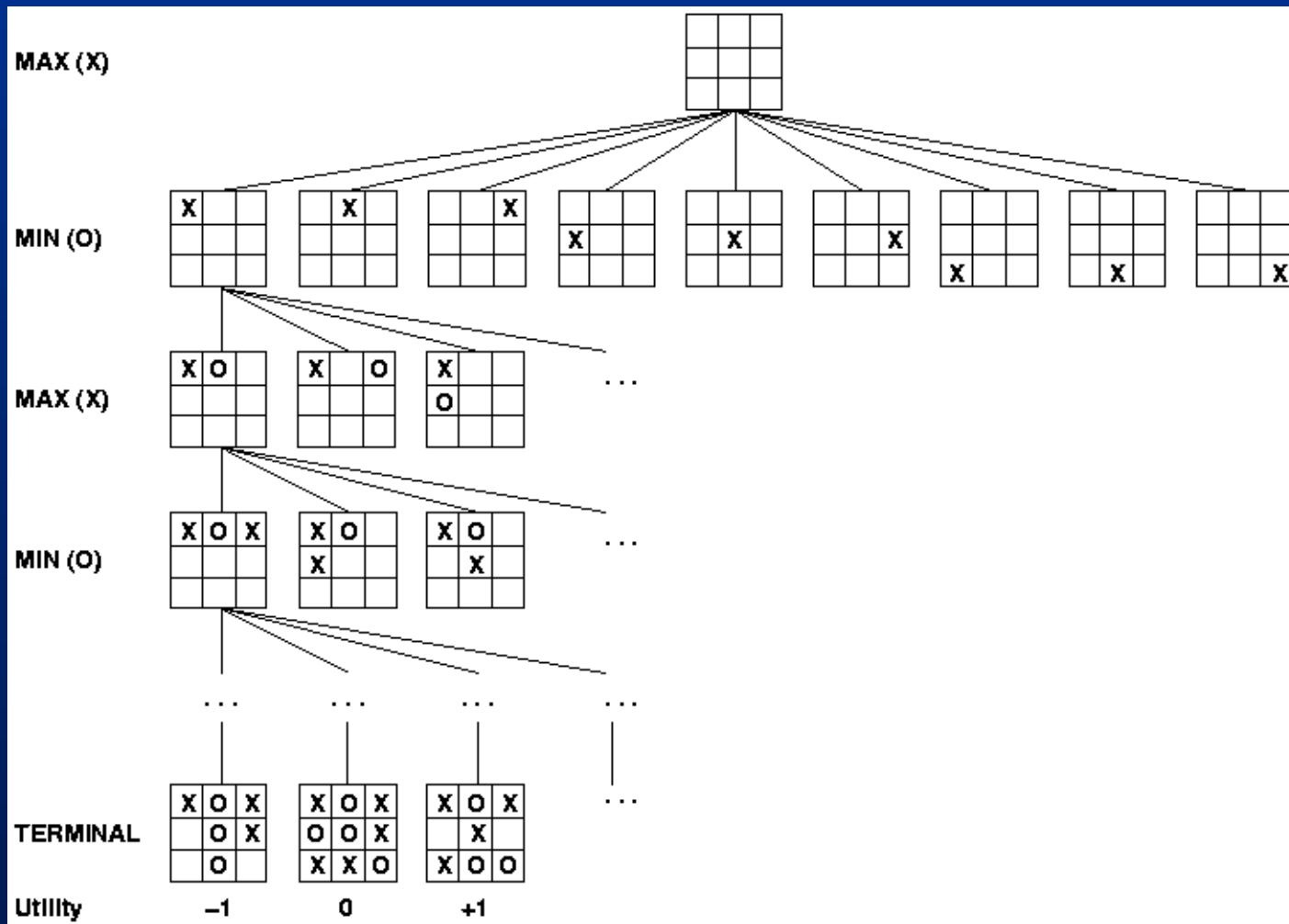
Którą drogę warto wybrać?

Oceniamy liście i cofamy się do góry przenosząc najwyższe lub najniższe oceny na węzły z ruchami dla MAX i MIN.

MAX powinien przejść do sytuacji B, skąd ma szansę dotrzeć do węzła  $W/+10$ , ale MIN zredukuje ją do  $W/+3$ ; wybór przejścia do C daje MIN szansę zredukowania uzyskania przewagi o 1 punkt, czyli  $L/-1$ .



# Kółko i krzyżyk



Strategia i oceny złożoności.

# Heurystyki oceny

- Koszty materialne: pionek =1, skoczek=3, goniec=3, wieża =5, hetman=9 ...
- Pozycja figur: dla każdej konfiguracji inne oceny, np. pionek w pobliżu końcowego pola jest dużo ważniejszy niż pionek zablokowany.
- Ocena następnego ruchu, zagrożeń dla poszczególnych figur.
- Złożone oceny konfiguracji wielu figur.
- Kombinacja liniowa różnych elementów oceny  $f_i$  z wagami  $W_i$

$$E(f) = \sum_i W_i f_i$$

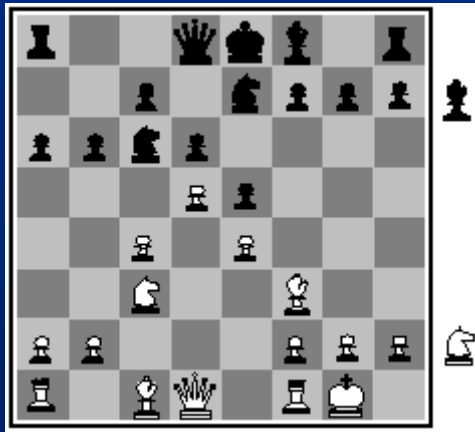
Współczynniki  $W_i$  dobiera się tak, by maksymalizować zyski.

Nieliniowe funkcje oceny mogą dać lepsze rezultaty – uczenie NN.

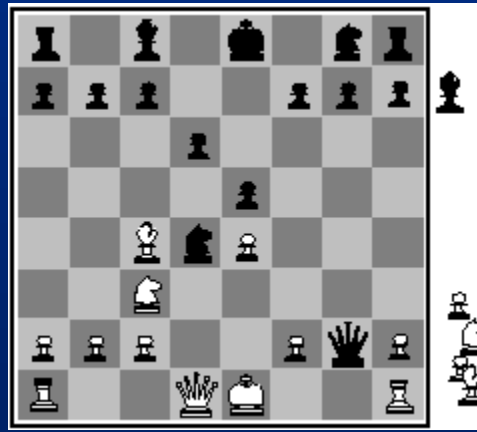
Precyzyjne wartości  $f_i$  nie mają znaczenia - liczy się tylko względny porządek, jest to więc „porządkowa funkcja kosztu”.

# Przykłady ocen

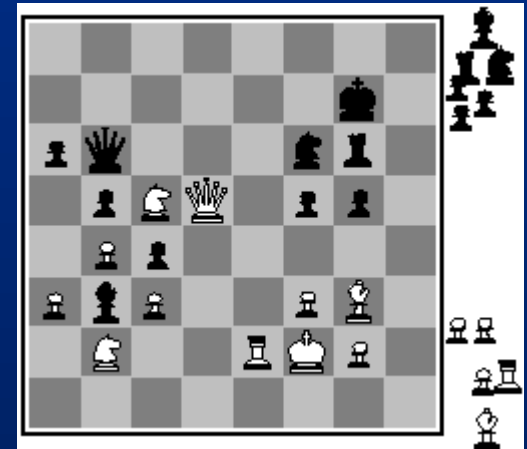
- Ludzie oceniają jakościowo, wystarczą względne porównania.
- Na podstawie doświadczenia tworzy się złożone funkcje oceny.



Ruch czarnych  
Przewaga białych



Ruch białych  
Czarne wygrywają



Ruch czarnych  
Białe bliskie przegranej



# Obcinanie alfa-beta

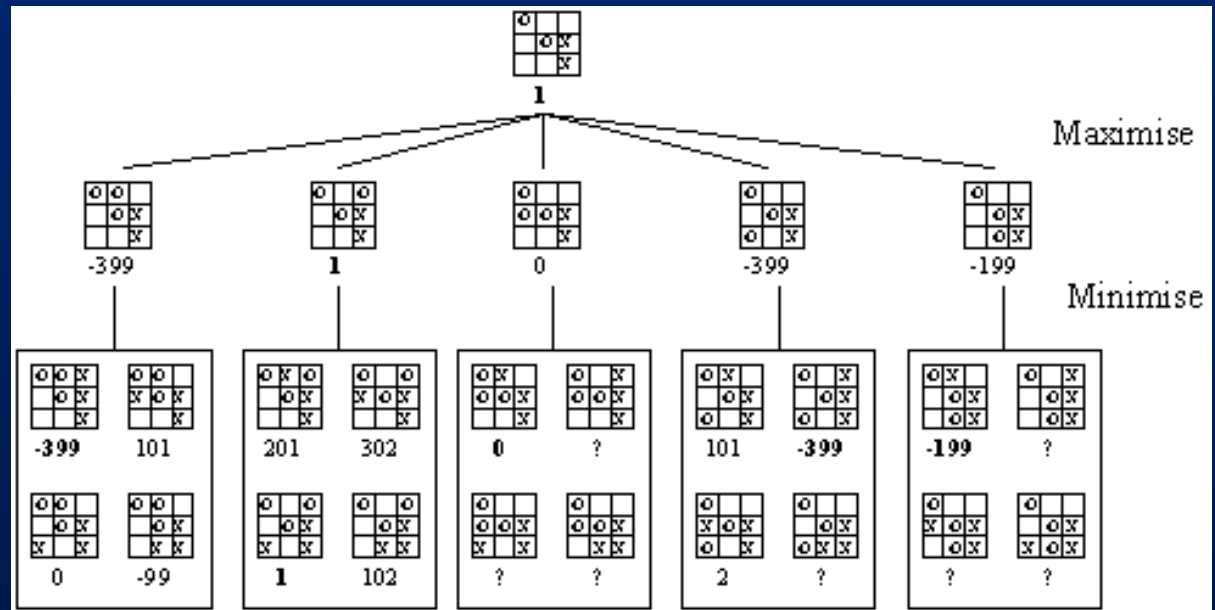
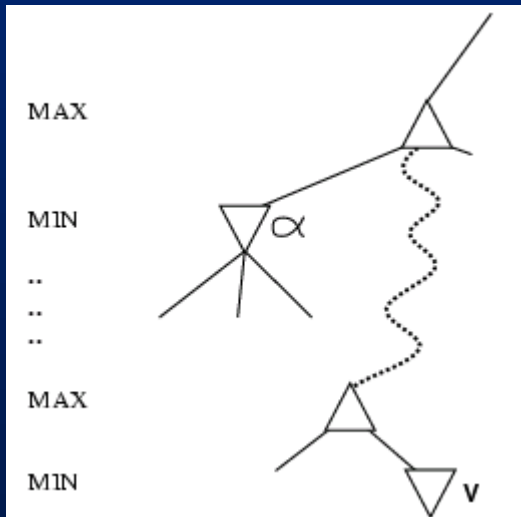
Technika pozwalająca oszczędzać czas w algorytmie min-max, obcinając gałęzie, które nie mają wpływu na końcowy wybór.

$\alpha$  - najlepsza wartość dla MAXa dla kolejnych kroków;

porzuć V jeśli masz lepsze wartości  $\alpha$  w innej części grafu (dla MAX);

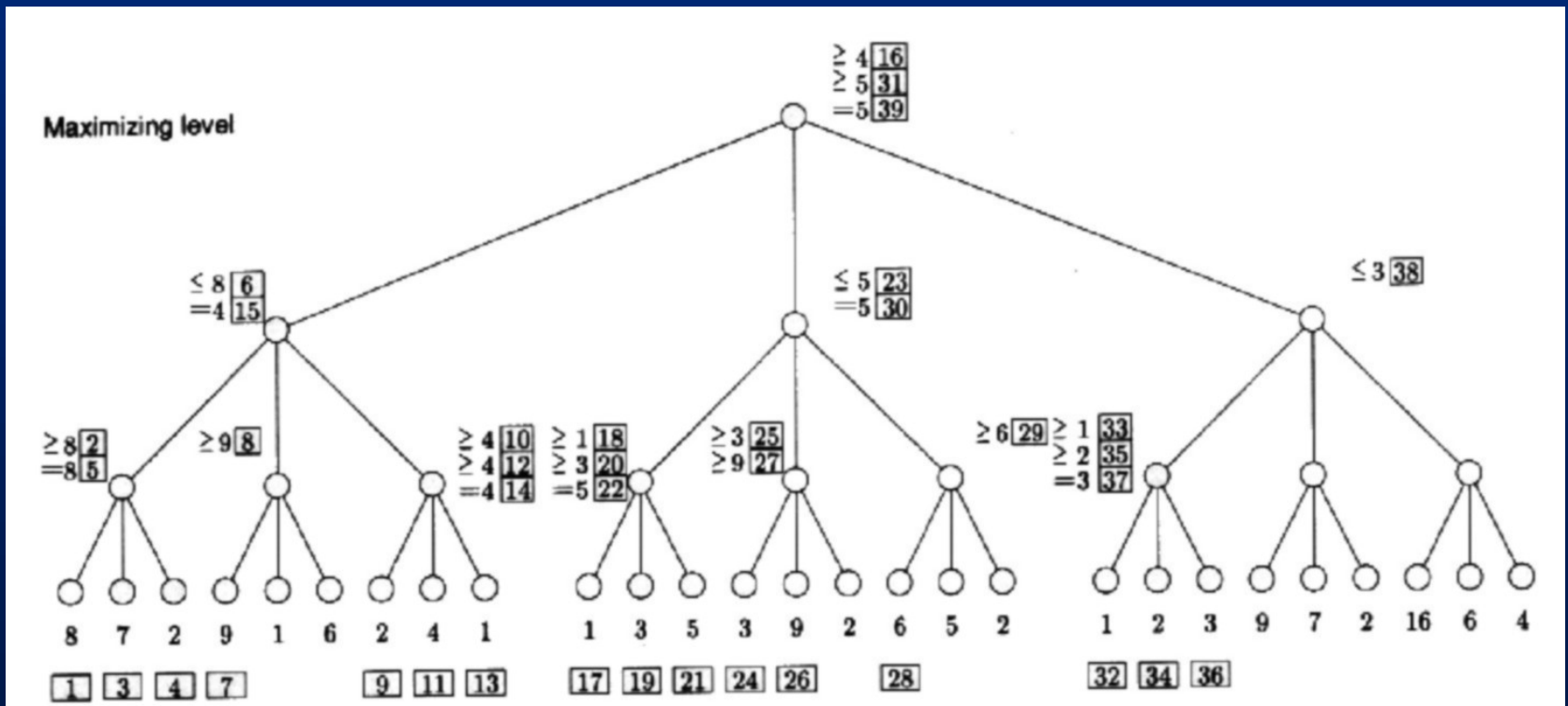
$\beta$  - najlepsza wartość dla MINa dla kolejnych kroków;

oszczędność w najlepszym razie  $t \sim O(b^{m/2})$



# Przykład $\alpha$ - $\beta$

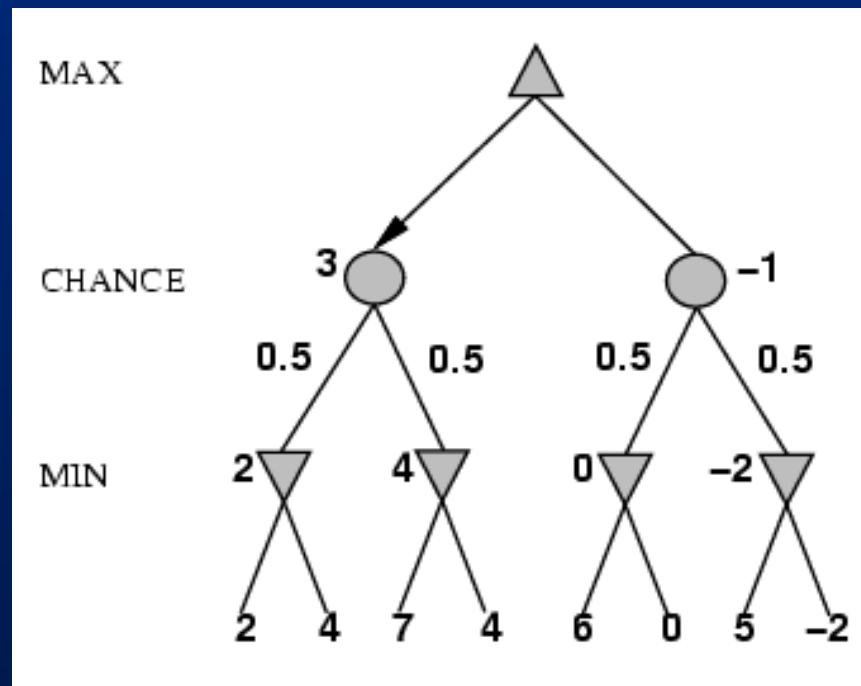
Kolejność odwiedzanych węzłów jest w kwadratach, a wartości f. heurystycznej obok. Alfa robi tu 2 ruchy, Beta 1.  
 W1 na najniższym poziomie ma  $f=8$ , więc węzeł 2 ma  $f \geq 8$ ; po węzłach 3 i 4 zostaje  $f=8$ ; dla Beta węzeł powyżej ma  $f \leq 8$ , po sprawdzeniu W7 jest  $f=9$ , więc pozostałe są pomijane.  
 W prawej gałęzi Alfa ma  $\max f=4$ , więc na tyle może liczyć MIN w W14.



# Gry niedeterministyczne

Strategia min-max połączona z oceną probabilistyczną szans na generowanie kolejnego ruchu.

Obcinanie  $\alpha$ - $\beta$  można stosować ale staje się znacznie mniej efektywne, wzrasta liczba możliwych rozgałęzień.



# Warcaby

- 1952, Samuel, pierwszy program uczący się gry w warcaby.
- 1992, Chinook (J. Schaeffer, UoA) wygrywa US. Open.  
Program używa szukania  $\alpha$ - $\beta$
- Mistrzostwa człowiek-maszyna, Londyn 1992  
Dr. Marion Tinsley, wygrał z Chinookiem 4-2, 33 remisy.  
Użyto 8-proc. stacji Silicon Graphics 4D/480, 40 Mhz, 256 MB RAM,  
128 Mflop. baza danych wszystkich końcówek z 1-7 figurami +  
prawie połowa wszystkich partii z 8 figurami.
- 1994, remis 1-1 i 18 remisów.
- 1995, wygrana Chinooka 1-0 i 31 remisów.  
Użyto komputera SGI z 512 MB RAM.
- W 2007 roku udowodniono, że najnowsza wersja Chinook nie może przegrać partii, warcaby uznano więc za w pełni rozwiązane.

# Trik-trak, Othello

- Trik-trak (backgammon), popularny w Japonii. Zawiera element niepewności (rzut kostką).
- 1980, program BKG wygrał raz z mistrzem świata.
- 1992, program Tesauro, techniki jak dla warcabów + uczenie się (sieć neuronowa) lepszych ocen, ranking wśród 3 najlepszych graczy.
- 1995, Logistello zwycięża mistrza świata Takeshi Murakami 6 do 0!

Program grał parę tygodni sam z sobą poprawiając swoje funkcje oceny heurystycznej.

Mistrzowskie rezultaty w wielu grach osiągnięto dopiero w latach 1990.

Moc obliczeniowa to warunek konieczny, ale nie wystarczający ...

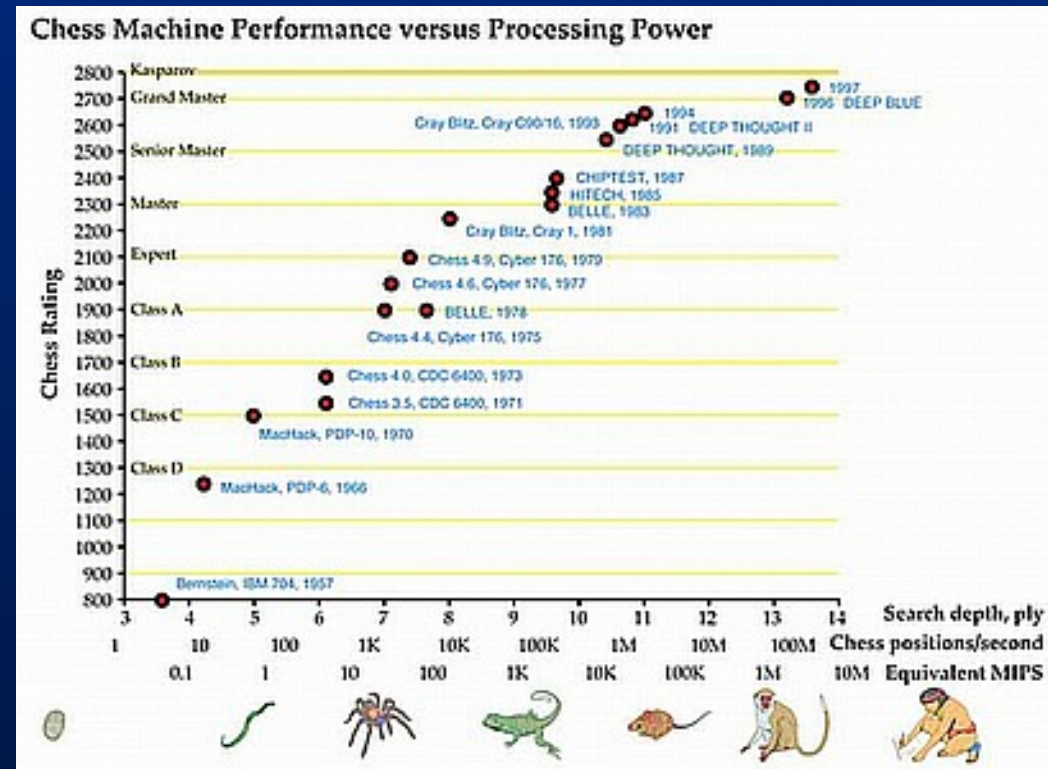
# Szachy – ogólnie

Statyczna i dynamiczna ocena sytuacji na planszy:  
liczba figur, wartość figur, położenie figur, możliwości ruchów.

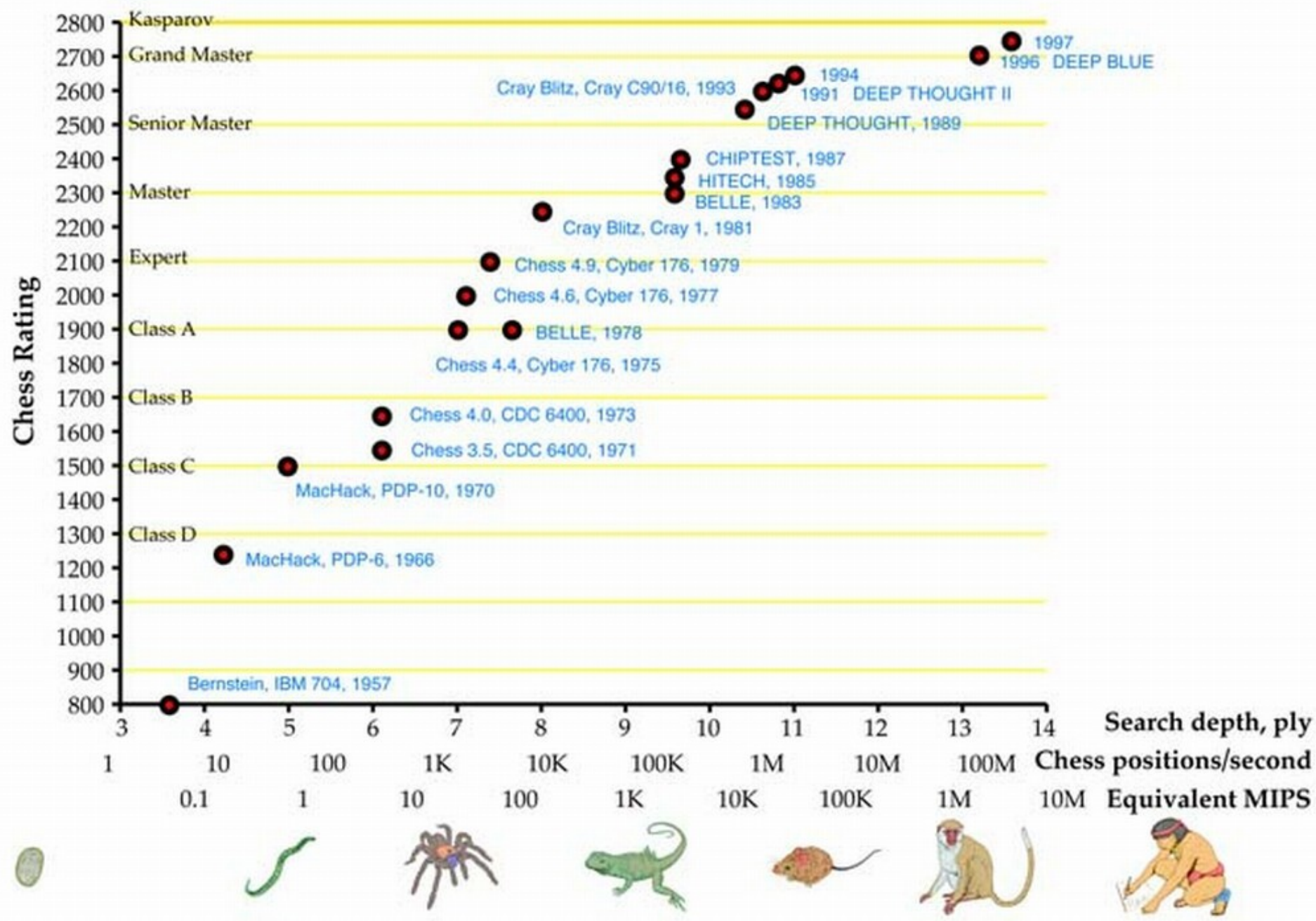
Heurystyczna funkcja oceny: suma  $W_i F_i$   
dobierz wagi  $W_i$  poszczególnych elementów oceny  $F_i$ .

Zależność:

głębokość szukania,  
liczba ocenianych pozycji,  
moc obliczeniowa w MIPS  
- siła programu mierzona  
w punktach ELO.

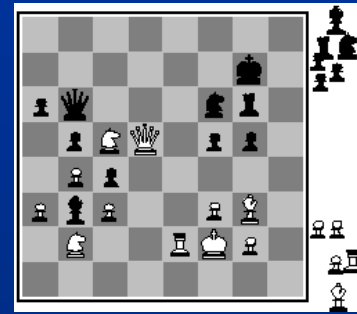


# Chess Machine Performance versus Processing Power





# Szachy – punkty



Mistrz świata > 2800 punktów. Przewidywana liczba ruchów:  
5 pełnych ruchów (10 poziomów) - 1500 punktów.

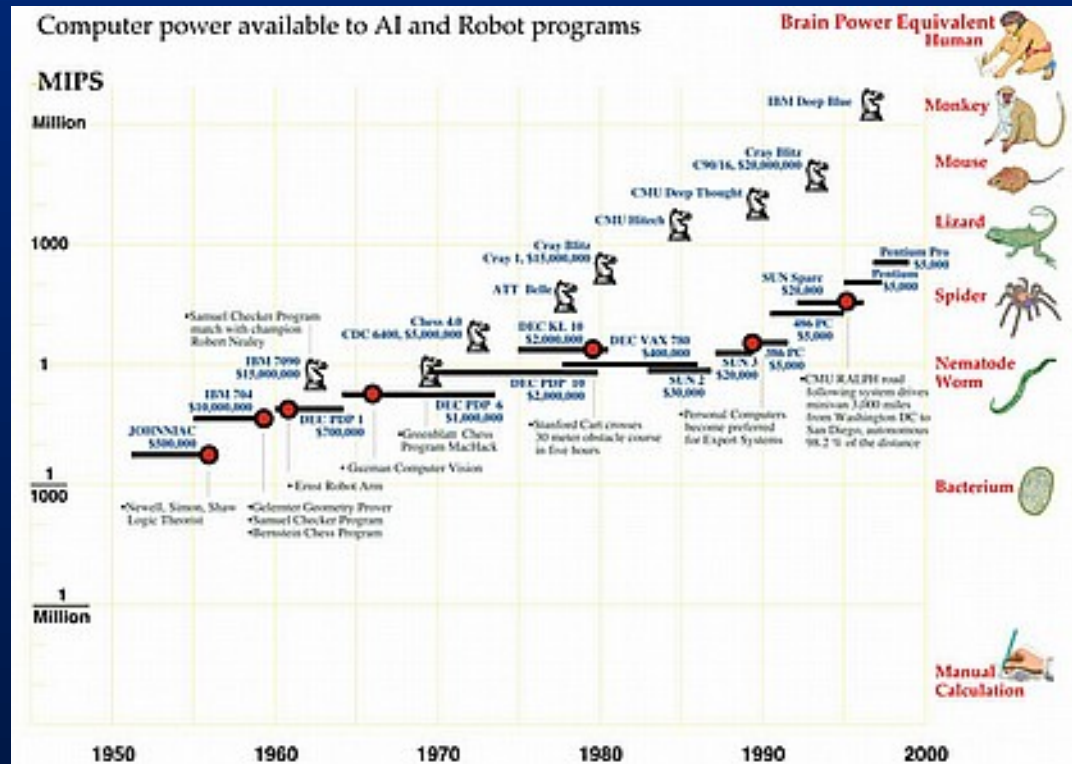
Od 5-10 poziomów mamy 200 p/poziom.

Dla 10 ruchów ok. 2500 punktów.

Ok. 35 ruchów/poziom, heurystyki redukują to do 6/poz, więc:  
dla 1000 ocen/sek, 150 sek/ruch,  $b=35$ , tylko ok. 3-4 ruchy - nowicjusz.

Potrzebna jest znacznie  
większa moc obliczeniowa!

Zależność  
jakość-szybkość obliczeń.  
MIPS = mln op/sek





# Computer power available to AI and Robot programs

Brain Power Equivalent Human

MIPS

Million

1000

1

1/1000

1 Million



Monkey



Mouse



Lizard



Spider



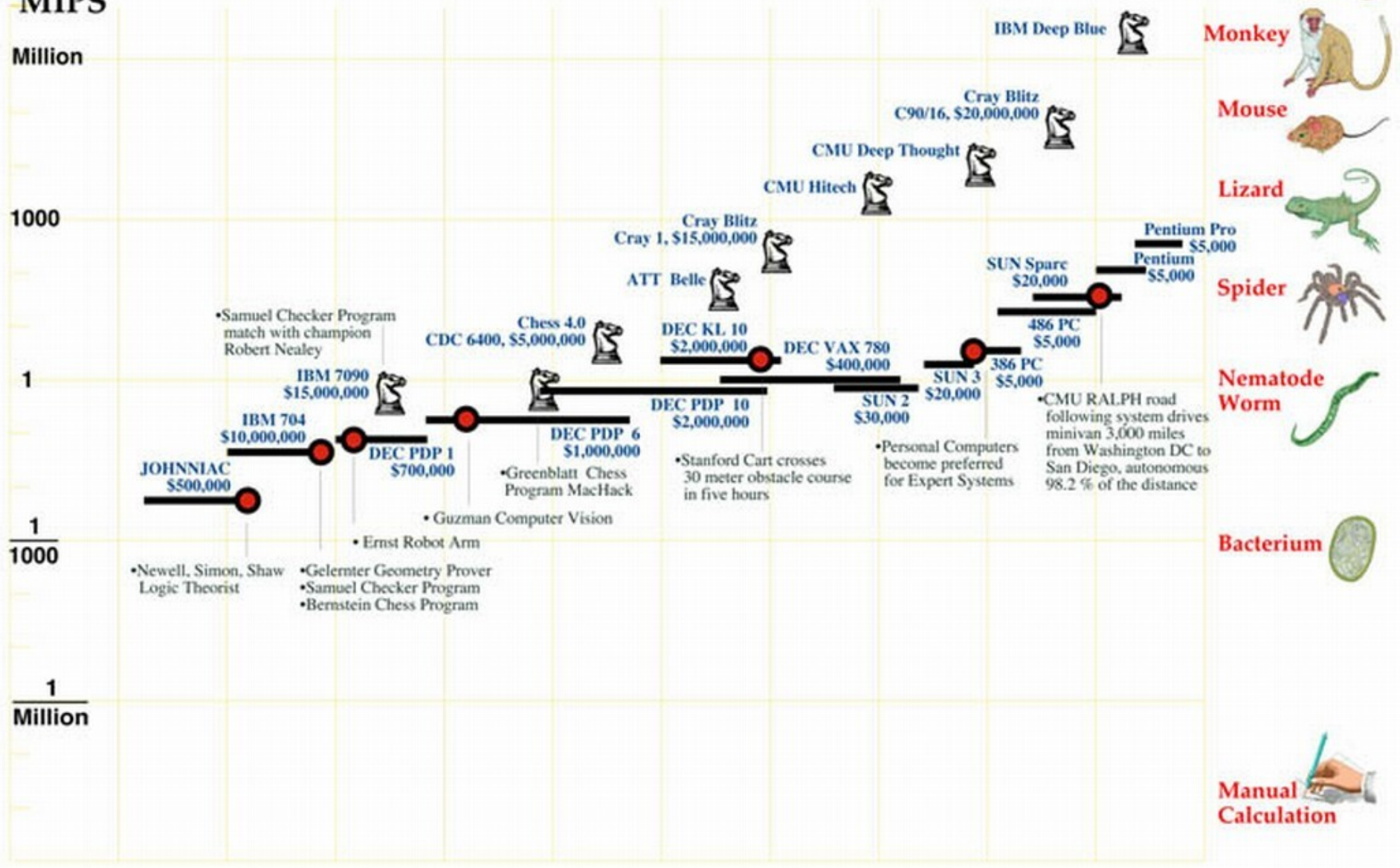
Nematode Worm



Bacterium



Manual Calculation



1950

1960

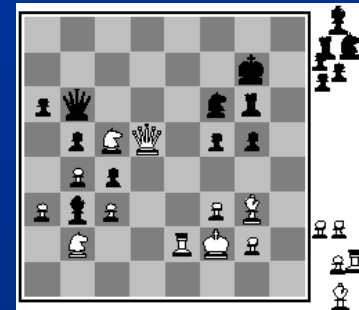
1970

1980

1990

2000

# Szachy – punkty



Mistrz świata > 2800 punktów. Przewidywana liczba ruchów:  
5 pełnych ruchów (10 poziomów) - 1500 punktów.

Od 5-10 poziomów mamy 200 p/poziom.

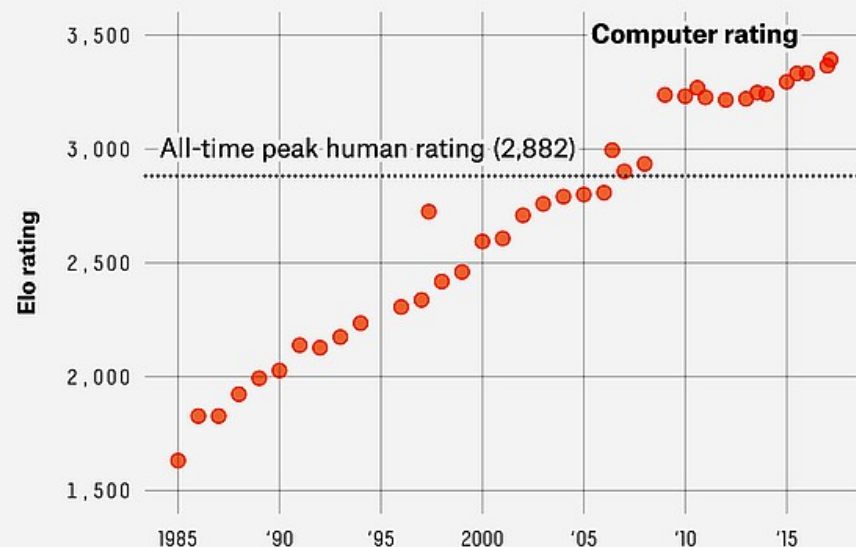
Dla 10 ruchów ok. 2500 punktów.

Ok. 35 ruchów/poziom, heurystyki redukują to do 6/poz, więc:  
dla 1000 ocen/sek, 150 sek/ruch,  $b=35$ , tylko ok. 3-4 ruchy - nowicjusz.

Potrzebna jest znacznie  
większa moc obliczeniowa!

Zależność  
jakość-szybkość obliczeń.  
MIPS = mln op/sek

**Chess computers have been better than humans for years**  
Elo ratings since 1984



# Szachy - historia.

- 1958, pierwszy program szachowy, Alex Bernstein.  
Szkocki międzynarodowy mistrz szachowy funduje nagrodę dla programu, który ogra go chociaż raz na cztery partie.  
W 1985 roku przegrał wszystkie cztery partie.
- 1985, HiTech wśród najlepszych 800 graczy, oceniał ok. 10 mln pozycji, w 1988 roku wygrał z arcymistrzem.  
Intel+IBM: szachy to dobra reklama.
- 1994 Chess Genius na PC Pentium, kilka razy zwyciężył Gary Kasparova; czas grania ograniczono do 25 minut na zawodnika.
- 1996 – Deep Blue przegrał z Kasparowem 2:3
- 1997 – Deep Blue wygrał 3.5:2.5
- 2002 – program Deep Fritz na PC remisuje z Vladimirem Kramnikiem  
Przez dwa miesiące Kramnik trenował z programem Deep Fritz.

# Deep Thought i Deep Blue

**Deep Thought**, od 1985 roku, 4 studentów z USA  
(T. Hsu, T. Anantharaman, M. Campbell, A. Nowatzyk).

Program **Deep Blue** (nowsze **Deep Thought**) + hardware do gry w szachy:  
32 procesory IBM RS6000/SP2 + 256 ASIC.

Ocena: ok. 200-1000 milionów pozycji/sek!  
Duża biblioteka otwarć i końcówek.

Deep Thought – szukanie z obcinaniem alfa-beta,  
przewiduje na ok. 10 ruchów w skomplikowanych sytuacjach.

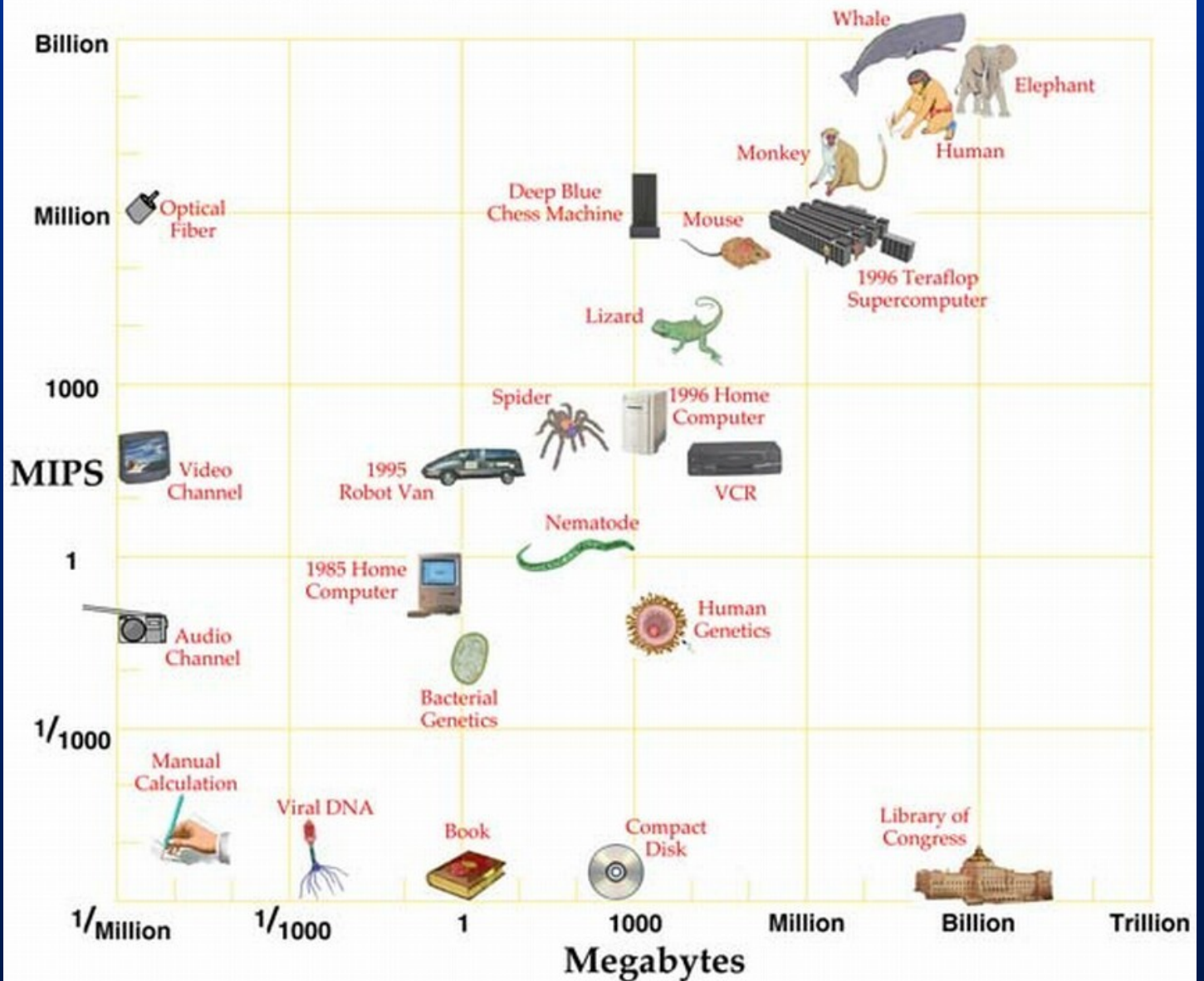
Deep Blue - ok. 14 ruchów, 3000 punktów, pobił Kasparova.

Reakcja prasy – potworna szybkość i pamięć zwyciężyły.

Mózg: 10.000 razy większa pamięć/szybkość więc to uwagi bez sensu!

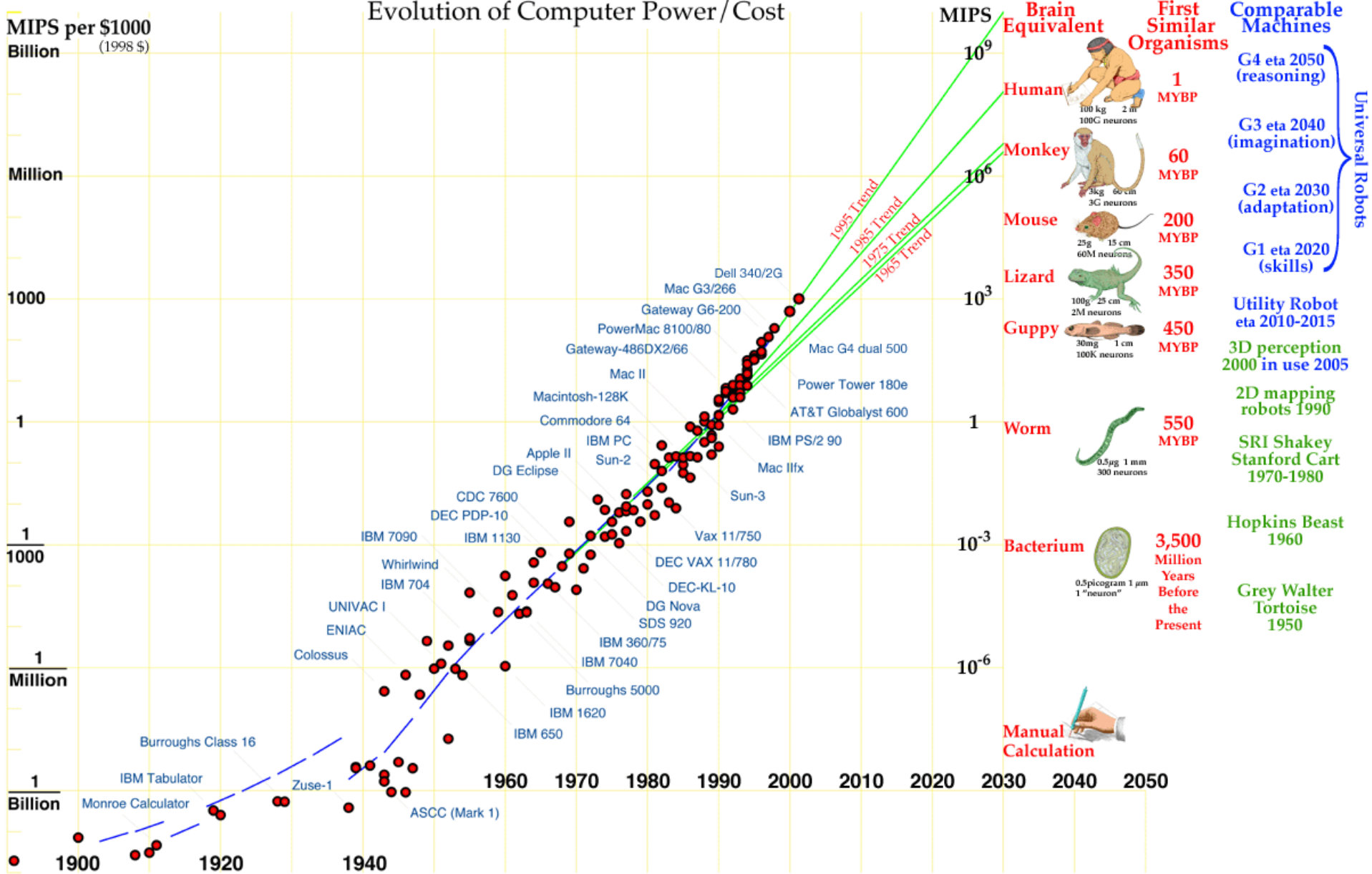
Moc obliczeniowa nadeszła i będzie coraz większa ...

# All Things, Great and Small





# Evolution of Computer Power/Cost



# Ostatni taki mecz ...

Kasparov vs. X3D Deep Fritz junior.

Nowy Jork, listopad 2003.

Mecz zakończył się remisem; główną atrakcją była gra w wirtualnej rzeczywistości.

Komputer, na którym zainstalowany był program Fritz był około 100 razy

wolniejszy niż Deep Blue i do tego zajęty generowaniem obrazu w 3 wymiarach, nic więc dziwnego, że mecz zakończył się remisem.

PC stoi w kącie ... następny taki mecz odbędzie się z programem działającym na telefonie?



Najwięcej punktów Elo w historii miał Carlsen 2882 i Kasparov 2851, tylko 4 osoby w historii miały ponad 2830 punktów.

Obecnie: >20 programów ma ELO>3200 punktów, jest kilka z ELO >3500.

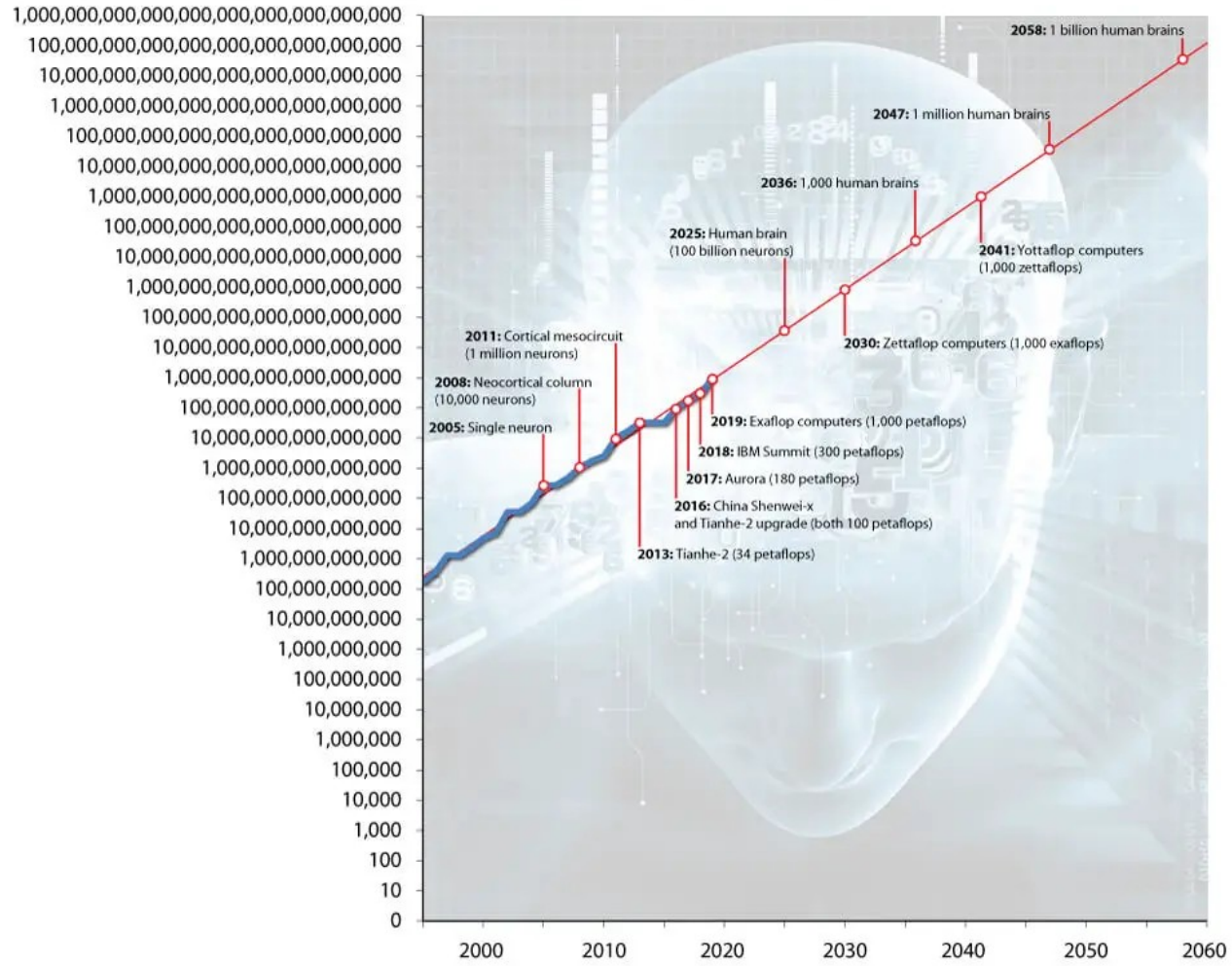
Jeden z najlepszych to Stockfish, program Open Source.

**Our world in data** - poziom ELO komputerowych programów szachowych.

# Eksponencjalny wzrost

Floating-point operations per second (FLOPS)

Exponential growth of supercomputing power, 1995-2060 (logarithmic scale)





# SuperMicro

SuperMicro to najmniejszy program szachowy, 455 bajtów!  
And do not think that will be so easy to defeat it!

```
#define f;if(
#define W while(
M=136,C=799,K=8,X,Y;char c[9],b[128]=
"VSUWTUSV";D(k,x,e,n){int i=0,j,t,p,u,r,y,m=
n?-C:e,v;do{f(u=b[i])&k}{j=" .H?LFICF"[p=r=u&
7]-64;W r=p>2&r<0?-r:64-"01/@AP@ABPOQ@NR_a@"
[++j]){y=i;do{t=b[y+=r]f(p==7|!*c)&&j==8|!(
r&7)-!t&p<3|t&k||y&M)break;v=t&k?8:
"#++;};Kk"[t&7]-35 f v>88)m=C f n){b[i]=0;b[
y]=u f p<3&&y+r+1&128)b[y]=(*c&&c[4]?c[4]:55
)-48|k,v+=M;v=-D(24-k,2,-e-v,n-1);f x&1){f v
+C&&i==X&y==Y){f j==8)b[y+(r>0?1:-2)]=0,b[(i
+y)/2]=6|k;return 0;}v=m;}b[i]=u;b[y]=t f v>
m){m=v f n>4)X=i,Y=y;}t+=p<5 f x&1&&(y&112
)+6*k==128&p<3)t--;}W!t);}W i=i+9&~M);
return m;}main(){X=8;W X--){b[X+112]=(b[X]-=
64)-8;b[X+16]=18;b[X+96]=9;}W 1){X=128;W X
--}putchar(X&8&&(X-=7)?10:" .?+nkbrq?*?NKBRQ"
[b[X]&15]);gets(c);X=*c-16*c[1]+C;Y=c[2]-16*
c[3]+C f!*c)D(K,0,0,5)f!D(K,1,0,1))K^=24;}]
```

# DeepChess

Nowa generacja programów szachowych uczy się sama.  
Niektóre programy nie potrzebują nawet reguł, bo rozpoznają je na podstawie obserwacji wielkiej liczby gier.

**Giraffe: Using Deep Reinforcement Learning** to Play Chess.

4-warstwowa sieć neuronowa, ocena setek milionów pozycji na szachownicy;  
bootstrap – program gra sam z sobą i na podstawie wyniku partii ocenia czy pozycja była silna czy słaba.

Gra na poziomie FIDE International Master na PC.

Jeszcze lepsze wyniki osiągnął program DeepChess: End-to-End Deep Neural Network for Automatic Learning in Chess. Technika jest podobna do Giraffe, program ocenia pozycje w „intuicyjny” sposób, ma styl podobny do profesjonalnych szachistów, poświęca dużo figur.

**Ten wykres** pokazuje obecny poziom komputerowych programów szachowych.  
**DeepLearning w szachach** - czy ogólne modele językowe, takie jak GPT, dobrze grają w szachy? Nie tak jak programy szachowe: GPT-3 konfabuluje i oszukuje, ale GPT-4 radzi sobie nieźle.

# Go: większe wyzwanie

Liczba ruchów w Go to średnio 260 (szachy tylko 35), a liczba różnych partii to  $10^{260}$  (szachy  $10^{123}$ ). Obydwie te liczby są bliskie  $\infty$ .

Liczba różnych pozycji na planszy:  $10^{172}$  (szachy  $10^{46}$ ).

Techniki szukania są mało przydatne: programy do go były długo słabe na standardowej planszy 19x19, ale na poziomie mistrzowskim na planszach 9x9.

Pierwszy program do gry w Go powstał w 1968 roku.

Ufundowano nagrodę 1 mln \$ dla programu, który pokona mistrza z Taiwanu!

MoGo (Many Faces of Go, Ver. 12) na komputerze 15Tflop, Monte Carlo Tree Search (MCTS), wygrał z mistrzem Myungwan Kim (8 Dan), przy handicapie 9 kamieni. Więcej rezultatów [komputerowego go](#).

Konieczne do dobrego grania w Go jest:

- rozpoznawanie struktur (typ lokalnych konfiguracji),
- reprezentacja relacji przestrzennych, uczenie maszynowe,
- strategie i planowanie, metody reprezentacje wiedzy.

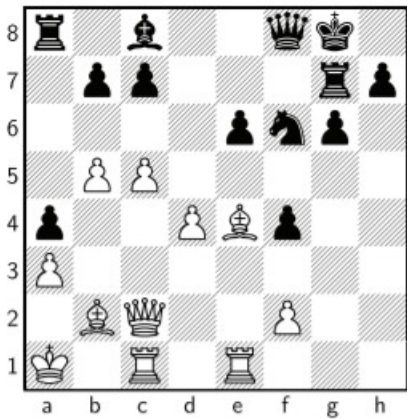
# AlphaGo

- Nowe techniki uczenia oparte na głębokich, wielowarstwowych sieciach neuronowych rozwijane są intensywnie znajdując wiele zastosowań i rozwiązując trudne problemy.
- Google AlphaGo zwyciężył pod koniec 2015 roku z Fan Hui, Europejskim mistrzem Go, 5 razy na 5 partii w warunkach takich jak na turniejach Go.
- Walka z mistrzami Go, Koreańczykiem Lee Sedol, odbyła się w marcu 2016; program wygrywa 4:1, w 2017 z mistrzem świata Ke Jie 3:0.
- AlphaGo jest uniwersalnym programem uczącym się, a nie wyspecjalizowanym programem do gry w Go. Program działał na komputerach z 48 jednostkami TPU (Tensor Processing Units).
- Program ma 4 elementy: algorytm szukania, symulacje Monte Carlo gier do oceny jakości ruchów; głęboką sieć neuronową (12 warstw), która nauczona została generowania ruchów na podstawie wzorów rozkładów pionków; drugą podobną sieć do oceny pozycji.
- Miesiące treningu na bibliotekach rzeczywistych gier, uczenie z nadzorem połączone z uczeniem z krytykiem.

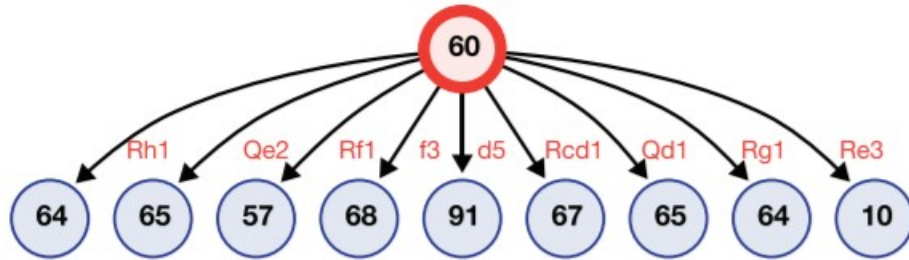
# AlphaGo Zero

- [AlphaGo Zero](#) opisany został jesienią 2017 roku, czyli niedługo po zwycięskim meczu z Lee Sidolem.
  - Programem jest uproszczeniem oryginalnego: nie ma tu ani procedur szukania ani Monte Carlo, jest tylko jedna ucząca się sieć neuronowa (samouczenie z krytykiem).
  - Program działa tylko z 4 TPU, na pojedynczym komputerze, nie korzysta z bibliotek ani wskazówek ekspertów.
  - Startując bez żadnej wiedzy o grze w Go (oprócz reguł grania) po 3 dniach grania ze sobą pobił oryginalny program AlphaGo 100:0.  
ELO Rating 5185, [highest human](#) Shin Jinseo (Korea) had 3864 points.
  - Program [Google DeepMind](#) nauczył się grać w 49 różne gry arkadowe.
- Transfer wiedzy pomiędzy różnymi grami jest jednak nadal słaby, to jedna z ostatnich barier by stworzyć uniwersalny system AI.

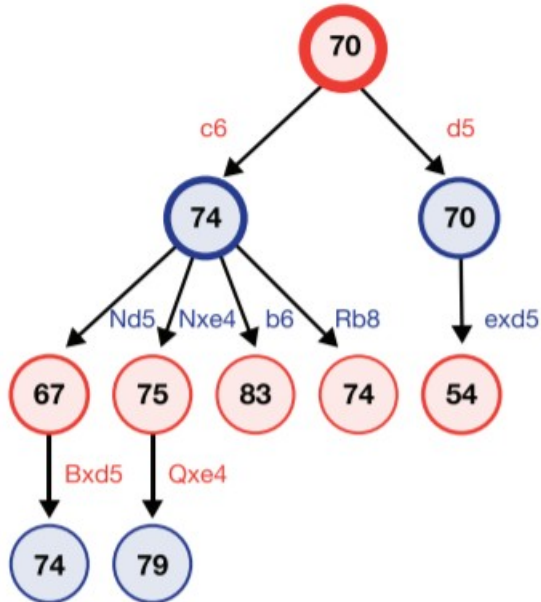




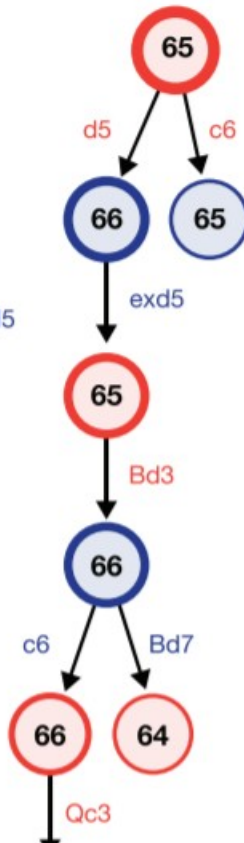
10<sup>2</sup> Simulations



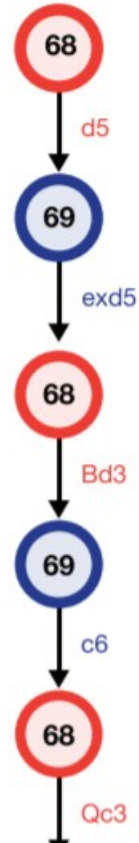
10<sup>3</sup> Simulations



10<sup>4</sup> Simulations



10<sup>5</sup> Simulations



10<sup>6</sup> Simulations



# AlphaGo zero czyli od 0 do mistrza!

Program gra ze swoją kopią i poporawia swoje funkcje heurystyczne

# Historia gier

## History of Game AI

By: Andrey Kurenkov

### Dartmouth Conference

1956: the birth of AI



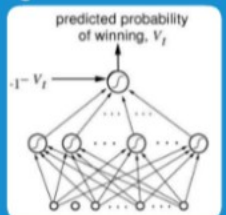
### Kaissa

1974: first world computer chess champion



### TD-Gammon

1992: RL and neural net based back-gammon AI shown



### Monte Carlo Go

1993: first research on Go with stochastic search

### NeuroGo

1996: ConvNet with RL for Go, 13 kyu (amateur)

### MCTS Go

2006: French researchers advance Go AI with MCTS

### Crazy Stone

2008: MCTS Go AI beats 4 dan player

### Zen19

2012: MCTS based Go AI reaches 5-dan rank

### Mac Hack

1967: chess AI beats person in tournament

### Samuel's Checkers AI

1956: IBM Checkers AI first demonstrated

### Zobrist's AI

1968: First Go AI, beats human amateur

### CNN

1989: convolutional nets first demonstrated

### CHINOOK

1994: checkers AI draws with world champion

### Deep Blue

1997: IBM chess AI beats world champion

### DeepMind

2014: Google buys deep-RL AI company for \$400Mil

### Bernstein's Chess AI

1958: first fully functional chess AI developed

### Checkers AI Wins

1962: Samuel's program wins game against person



### Backprop

1986: multi-layer neural net approach widely known



### AlphaGo

2016: Deep Learning+MCST Go AI beats top human





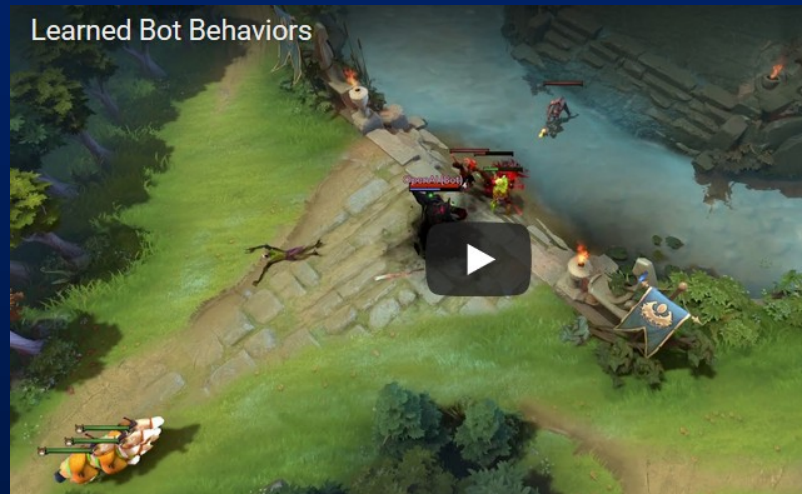
# Dota 2

Reguły w Go czy szachach są nieliczne i stosunkowo proste.

Dota 2 to gra wymagająca planowania, oszukiwania, ataku, zrozumienia intencji i planów przeciwników. Możliwości wyboru postaci i przedmiotów, którymi można się posługiwać są w zasadzie nieograniczone.

Program grupy OpenAI zwyciężył w listopadzie 2017 w turnieju z profesjonalnym graczem, mistrzem Dota 2.

Boty sterowane przez program uczyły się grając ze sobą przewidywać ruchy przeciwnika, działać w nowych sytuacjach, współpracować ze sobą. Kolejne wyzwanie to pełna gra 5 botów na 5 graczy.



# StarCraft II

StarCraft II to jedna z najbardziej popularnych gier typu RTS (real-time strategy), w której ważna jest strategia długoterminowa. Pełna informacja o sytuacji nie jest tu dostępna Szybkość maszyny została ograniczona by dać ludzkim zawodnikom szansę. W styczniu 2019 AlphaStar (DeepMind) zwyciężyła z dwoma najlepszymi profesjonalistami 5:0. A byli tacy pewni zwycięstwa ...

Głęboka sieć neuronowa trenowana była na danych bezpośrednio z gry, używając uczenia z nadzorem i z krytykiem.

AlphaStar używała strategii i technik walki nie widzianych nigdy u ludzi.

