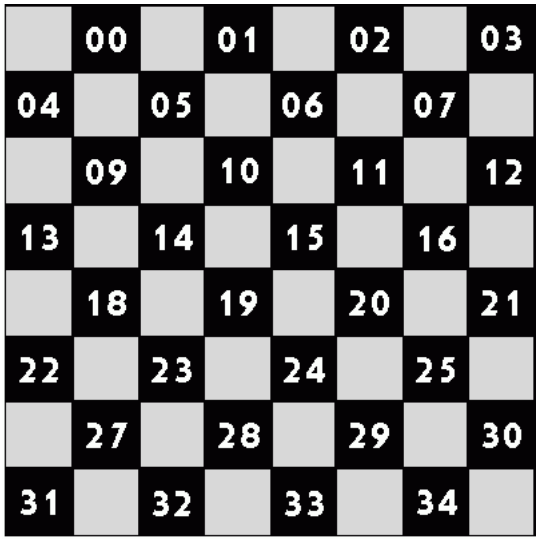
Zastosowanie metody Samuela doboru współczynników funkcji oceniającej w programie grającym w anty-warcaby

Daniel Osman

promotor: dr hab. inż. Jacek Mańdziuk

Spis treści

Anty-warcaby



- Zasady jak w warcabach
- Celem jest pozbycie się własnych pionków

Algorytmy przeszukiwania drzewa gry

- Minimax
- Alfa-beta
- Tablica transpozycji
- Heurystyka historyczna
- Iteracyjne pogłębianie
- MTD(f), MTD-bi

Alfa-beta

- Ulepszenie minimax
- Liczba wierzchołków w drzewie gry
  - maksymalnie

$$w^d$$

  - minimalnie

$$w^{\lfloor d/2 \rfloor} + w^{\lceil d/2 \rceil}$$

Tablica transpozycji

- Służy do zapamiętywania:
  - oceny stanu gry
  - dokładności tej oceny
  - najlepszego ruchu dla tego stanu
- Implementacja w postaci tablicy haszującej

Heurystyka historyczna

- Oceniać ruchy niezależnie od stanu
- Wykorzystywać oceny do sortowania kolejności odwiedzania następników
- Możliwych ruchów jest nie więcej niż 32 x 32 = 1024

Iteracyjne pogłębianie

- Przeszukiwanie drzewa gry kolejno na głębokość

$$d_0, \quad (d_0 + k), \quad (d_0 + 2k), \quad (d_0 + 3k), \quad \ldots$$

- Można zatrzymać obliczenia w dowolnym momencie
- Lepsze posortowanie dzięki tablicy transpozycji

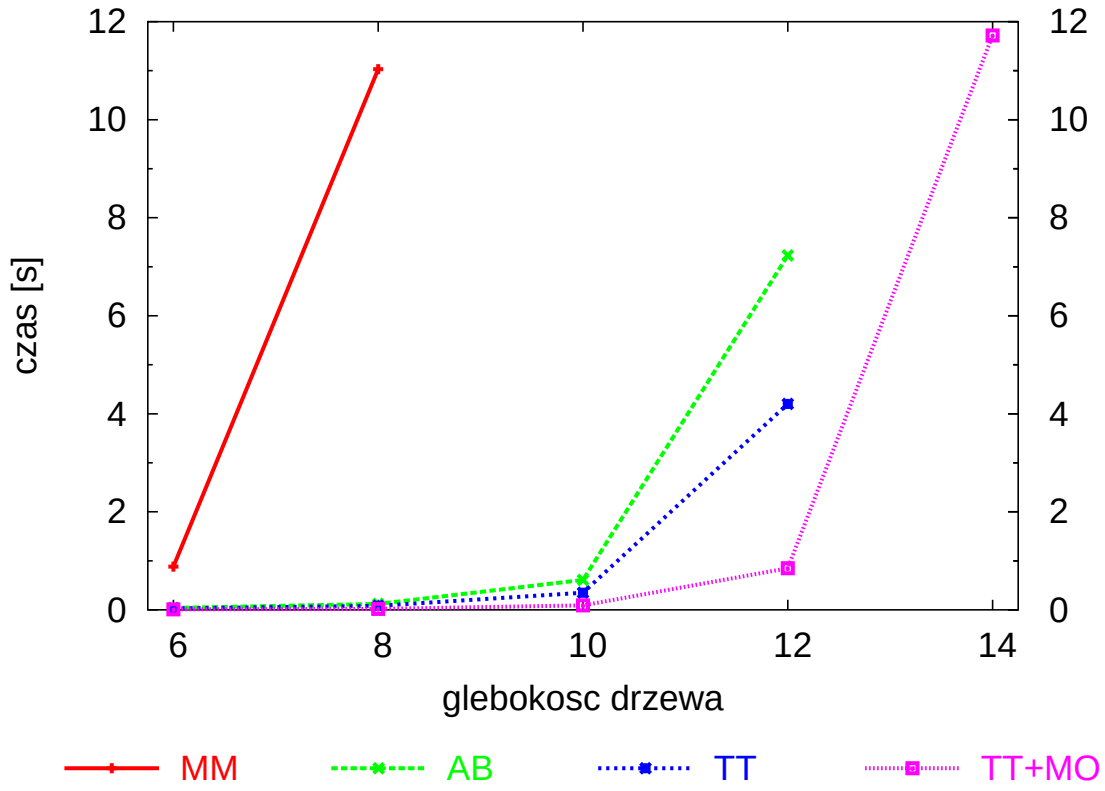
MTD(f)

- Funkcja MT: (Memory enhanced Test)
  - alfa-beta z zerowym oknem
  - szybkie ustalenie ograniczenia górnego albo dolnego
- Funkcja $MTD(f)$
  - wielokrotne wywoływanie funkcji MT w celu ustalenia szukanej wartości
  - parametr f: przybliżenie początkowe
- Najszybszy znany dzisiaj algorytm przeszukiwania drzewa gry

MTD-bi

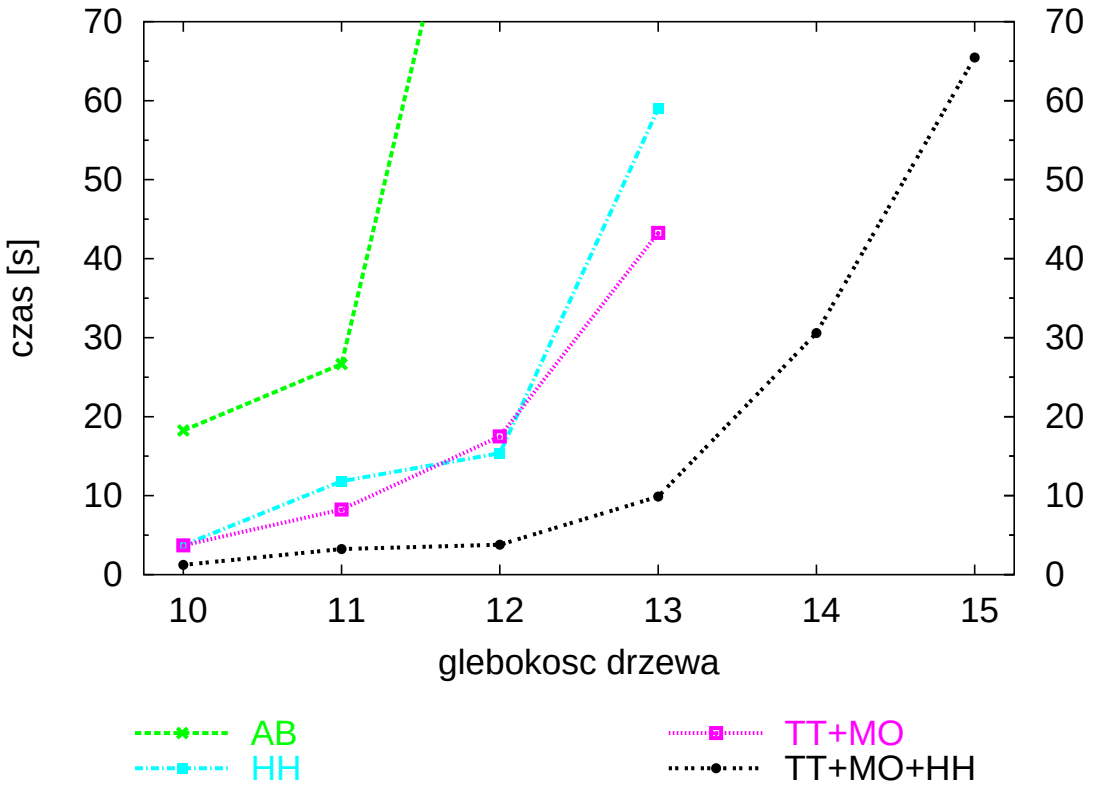
- Bisekcja
- Zalety
  - prosty, szybki i tak samo dokładny jak minimax
  - niepotrzebne przybliżenie początkowe
  - można stosować z ciągłą funkcją oceny
- Wady
  - MTD-step

Wyniki



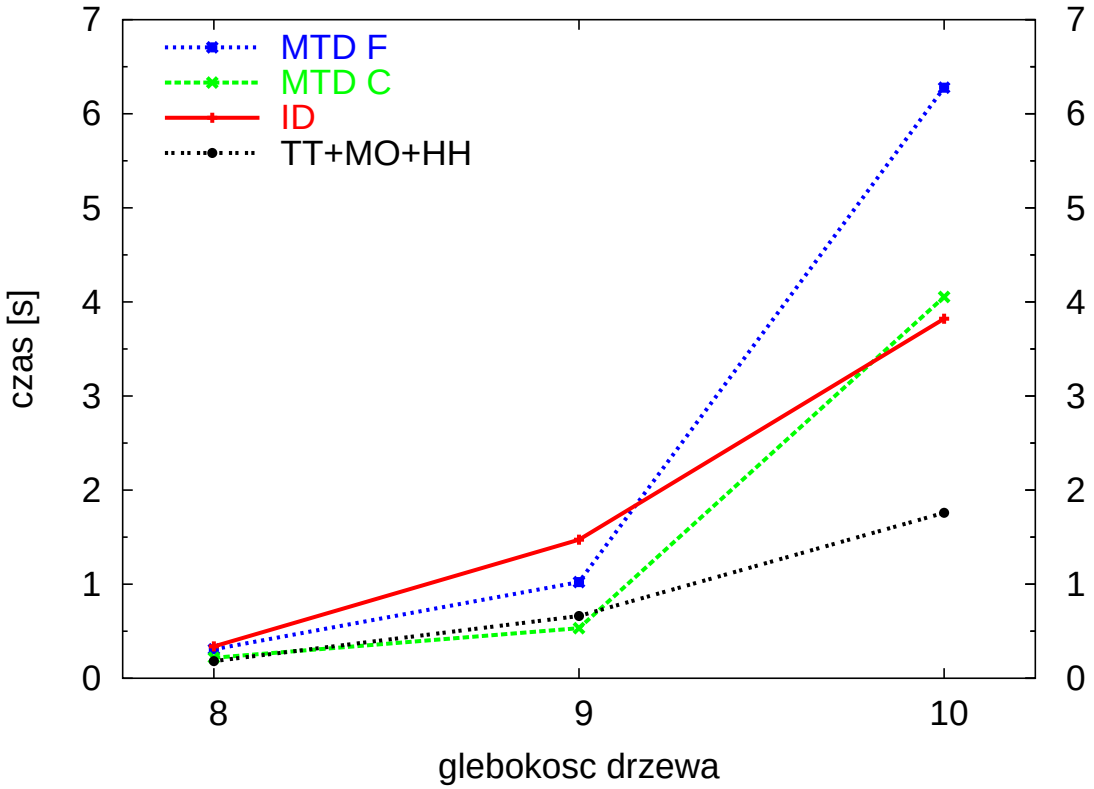
Średni czas oczekiwania na ruch w zależności od głębokości drzewa

Wyniki



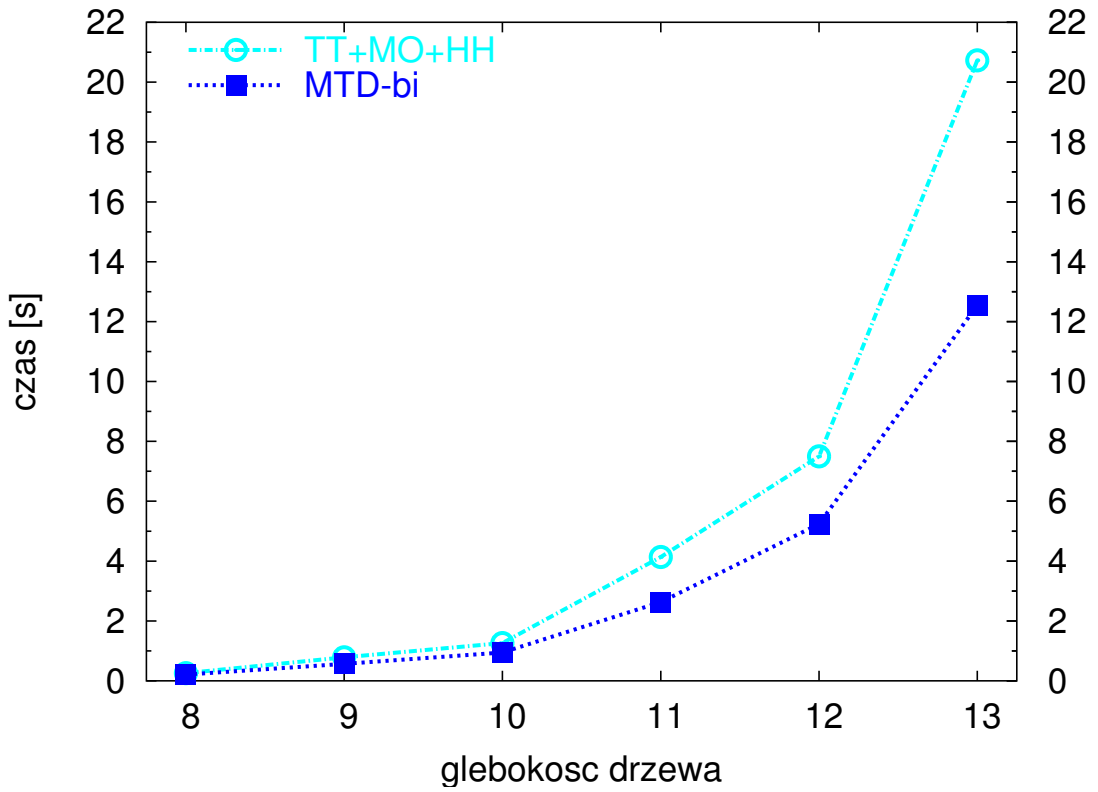
Średni czas oczekiwania na ruch w zależności od głębokości drzewa

Wyniki



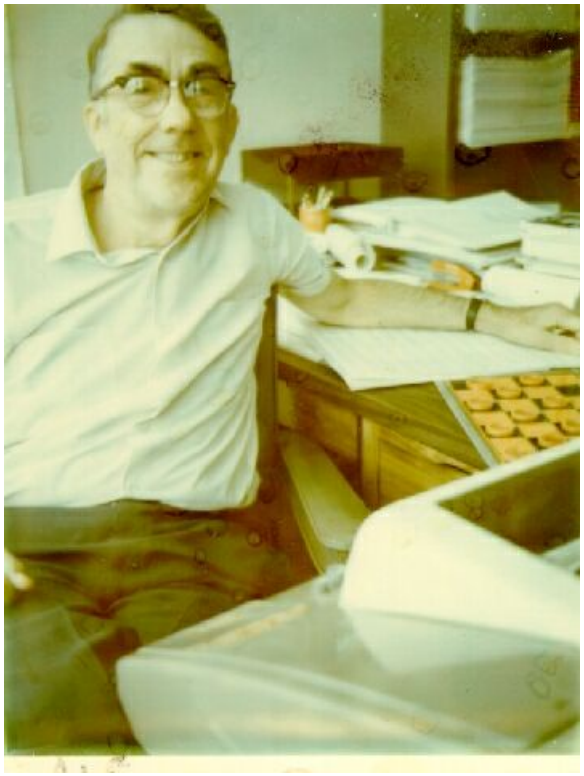
Średni czas oczekiwania na ruch w zależności od głębokości drzewa

Wyniki



Średni czas oczekiwania na ruch w zależności od głębokości drzewa

Uczenie



Arthur L. Samuel

Funkcja oceniająca

- Ocena stanu s:

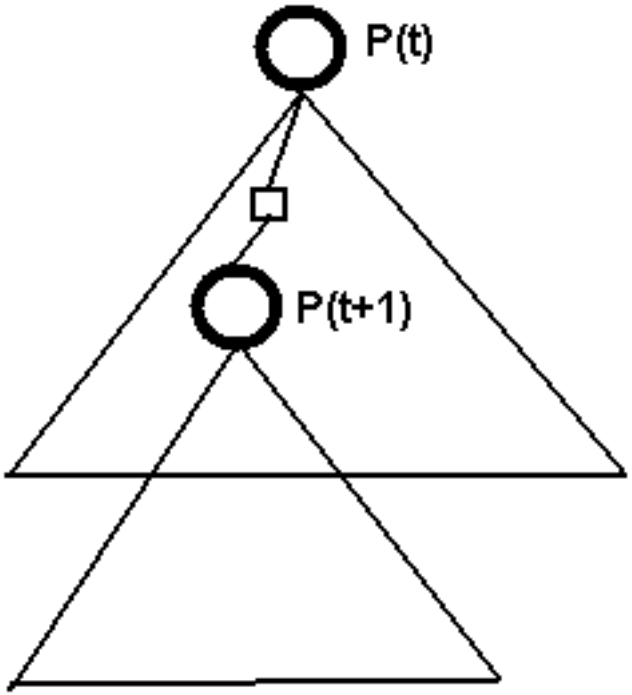
$$S(s, w) = w_1 \cdot x_1(s) + w_2 \cdot x_2(s) + \ldots + w_n \cdot x_n(s)$$

x_i - heurystyki Samuela

w_i - wagi funkcji oceniającej

- Uczenie polega na doborze wag w_i

Metoda Samuela



- Błąd funkcji oceniającej: $P^{(t+1)} - P^{(t)}$

Temporal difference learning

- Algorytm $TD(\lambda)$

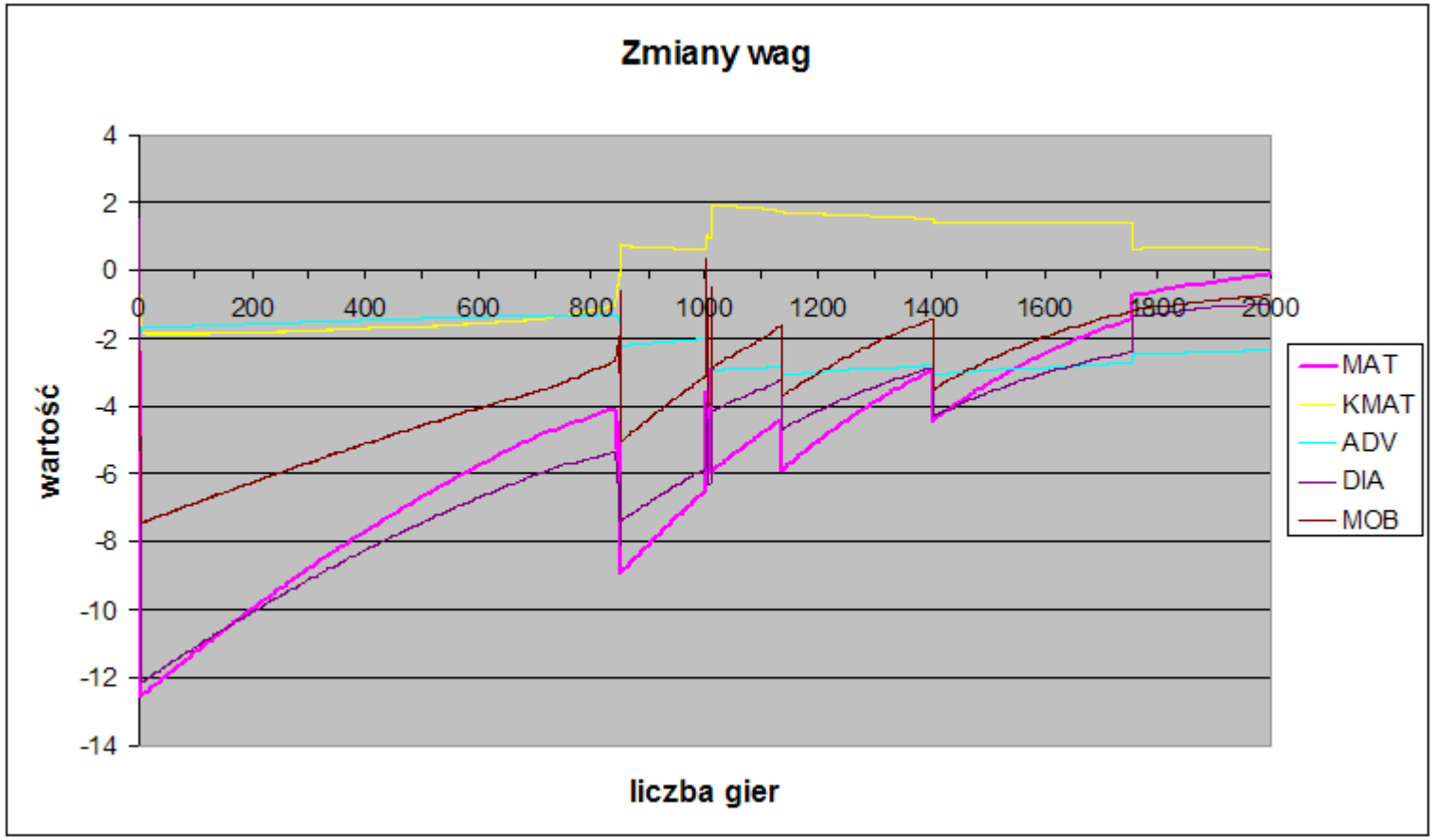
$$\Delta w^{(t)} = \alpha\left(P^{(t+1)} - P^{(t)}\right)\sum_{i=1}^{t}\lambda^{t-i}\nabla_w P^{(i)}$$

  - dla $\lambda = 1$: metoda Widrowa-Hoffa
  - dla $\lambda = 0$: metoda Samuela

Opis przeprowadzonych doświadczeń

- Gracz uczący się
- Przeciwnicy
- Głębokość przeszukiwania drzewa gry: $d = 4$

Problem z remisami



Problem z remisami

- Zastosowane rozwiązania:
  - pełna implementacja remisów
  - gra z wieloma przeciwnikami
- Wzrost skuteczności uczenia

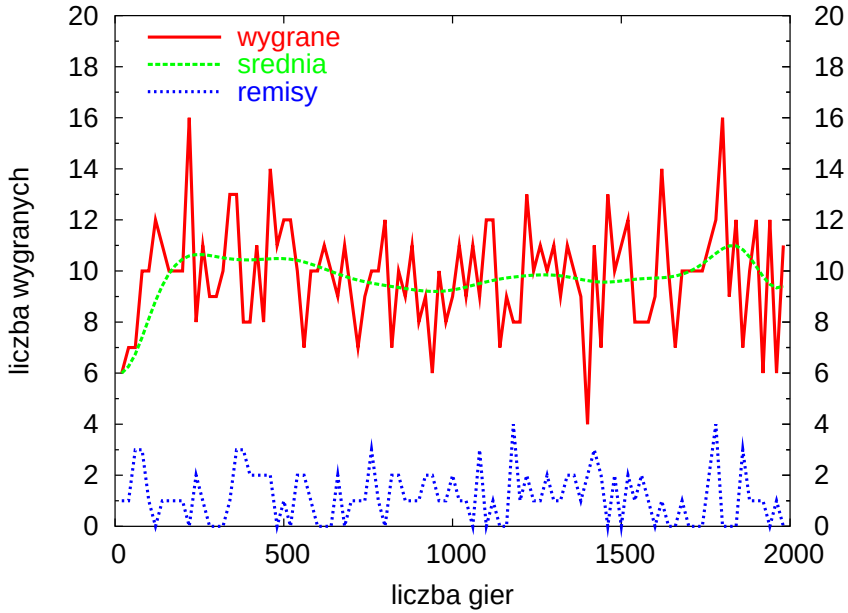
Grupa treningowa - grupa testowa

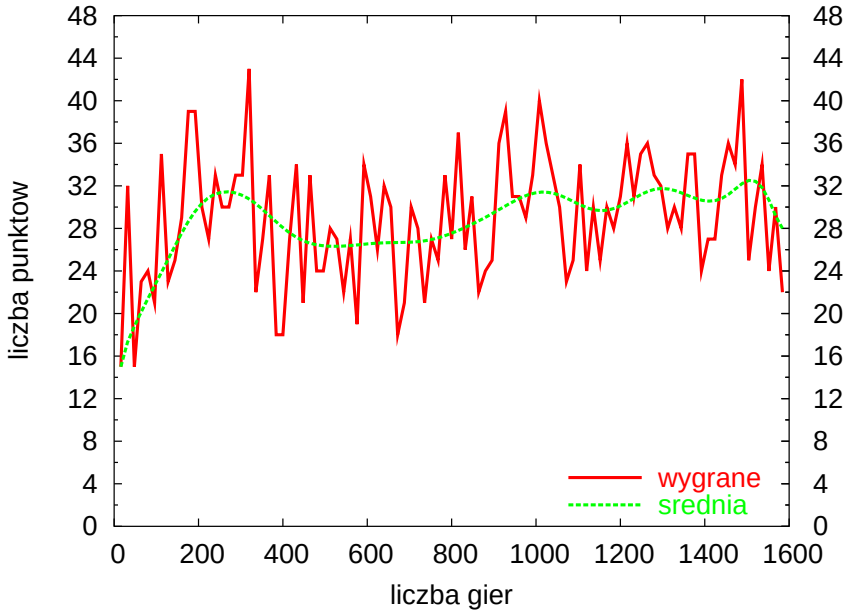
- Trening (zdobywanie doświadczenia)
  - jak gracz uczący się radzi sobie z bezpośrednimi przeciwnikami
- Testowanie (możliwość generalizacji)
  - jak gracz uczący się radzi sobie z nowymi przeciwnikami

Parametry α i λ

- Faza I
  - $\alpha = 1.0e^{-4}$
  - $\lambda = 0.95$
- Faza II
  - $\alpha = 1.0e^{-5}$
  - $\lambda = 0.7$
- Wynik zgodny z rezultatami innych badaczy

TD(λ)



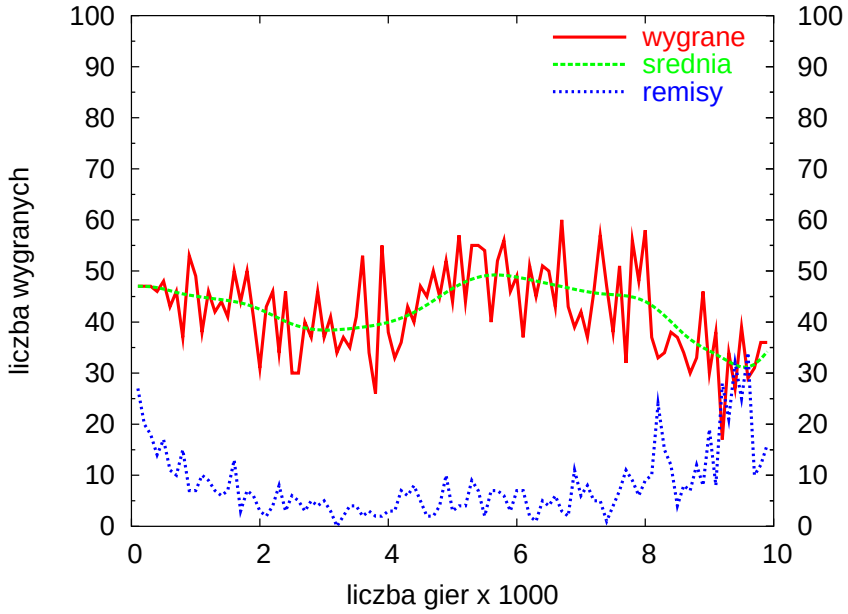


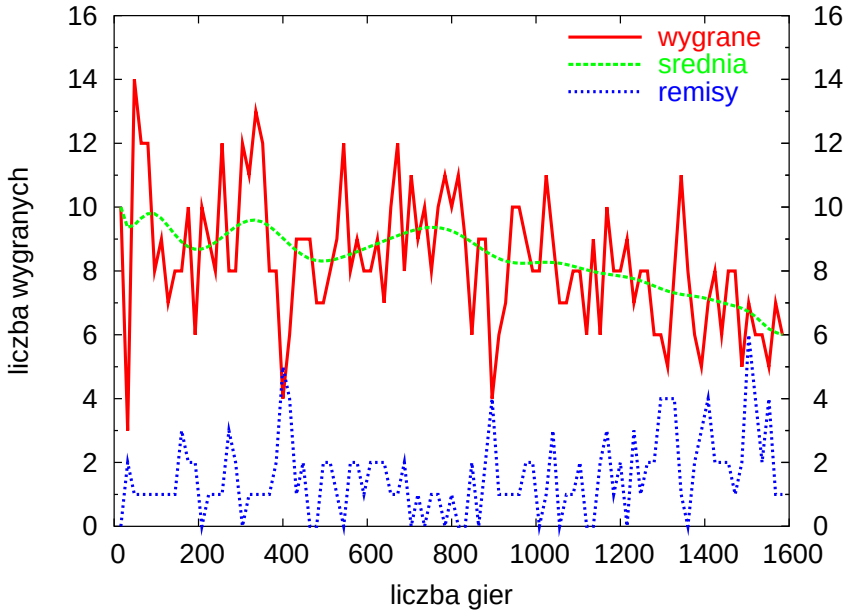
Liczba wygranych gracza uczącego się w grupie treningowej (20 przeciwników) oraz liczba zdobytych punktów w grupie testowej (16 przeciwników).

Metody treningu

- Skład grupy treningowej
  - liczba przeciwników
  - jakość przeciwników
- Czy zmieniać przeciwników?
- Jak długo trenować?

Metody treningu



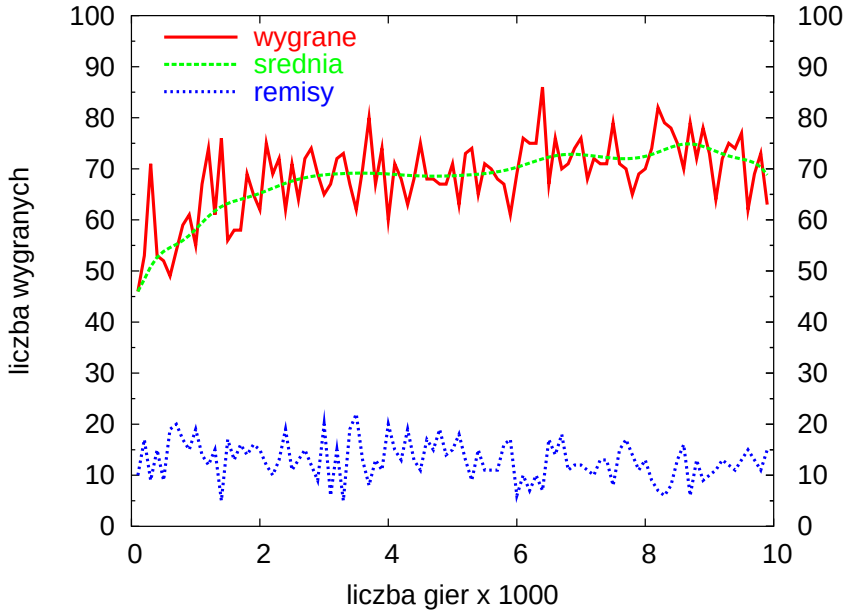


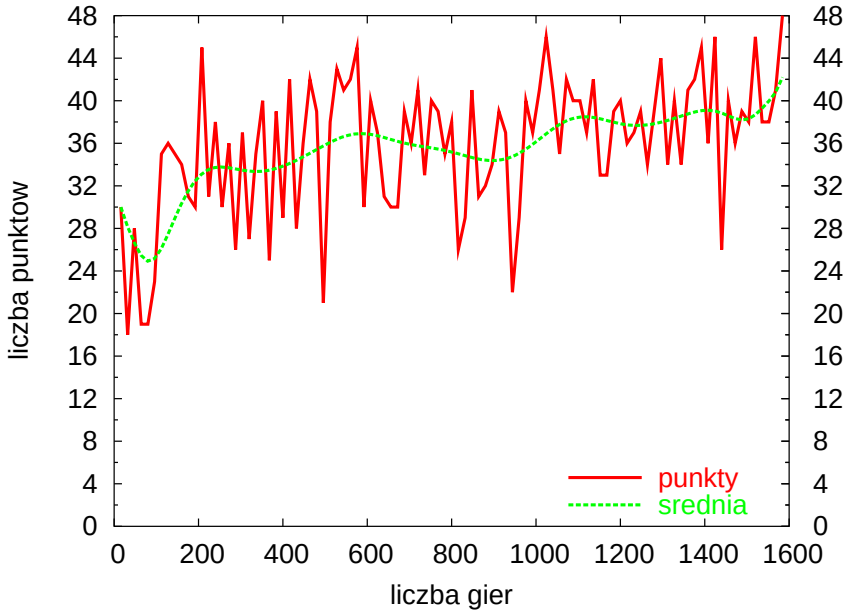
Zastosowanie mało efektywnej metody treningu

Ulepszenia

- Uczenie na porażkach
- Algorytm TDLeaf
- Uczenie na ruchach przewidzianych przez gracza uczącego się

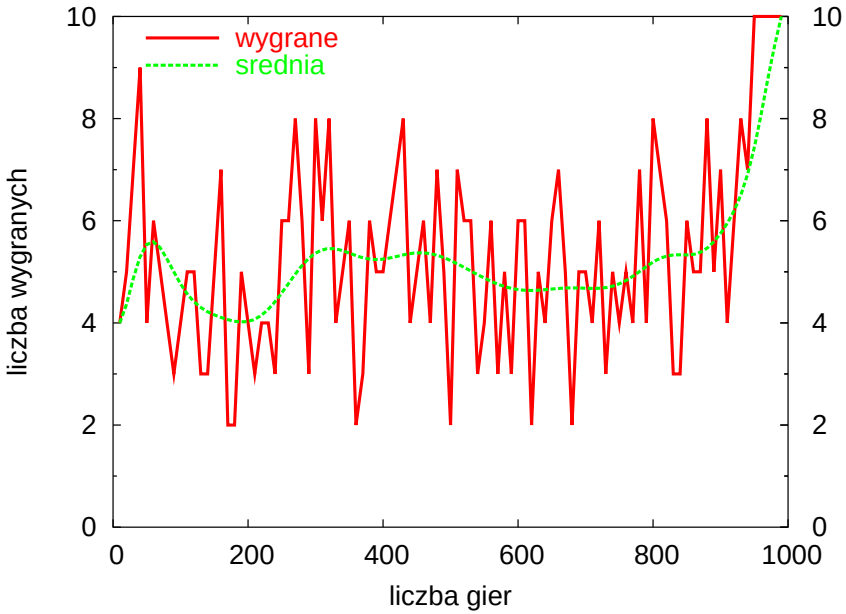
Ulepszenia

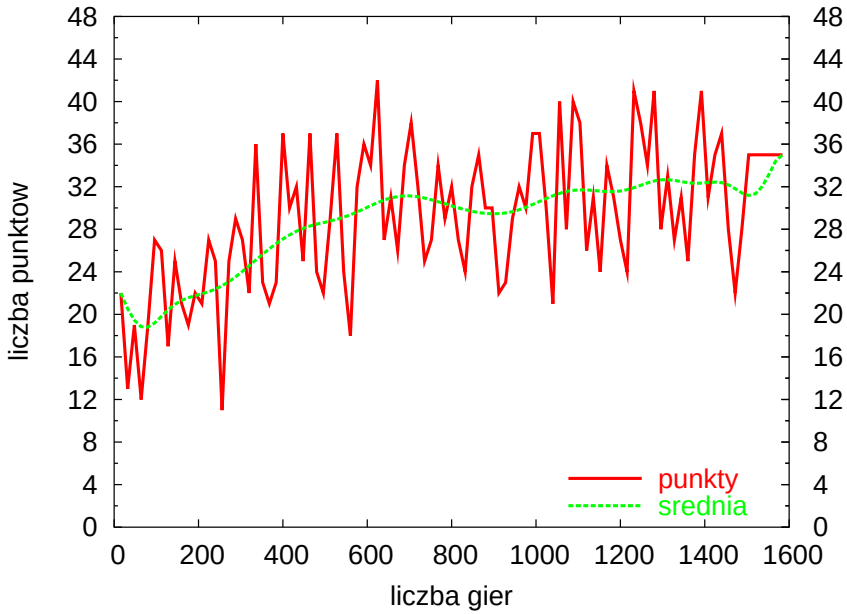




Najlepszy uzyskany wynik: trening na grupie 10 zróżnicowanych jakościowo przeciwnikach, 10,000 gier, uczenie na porażkach.

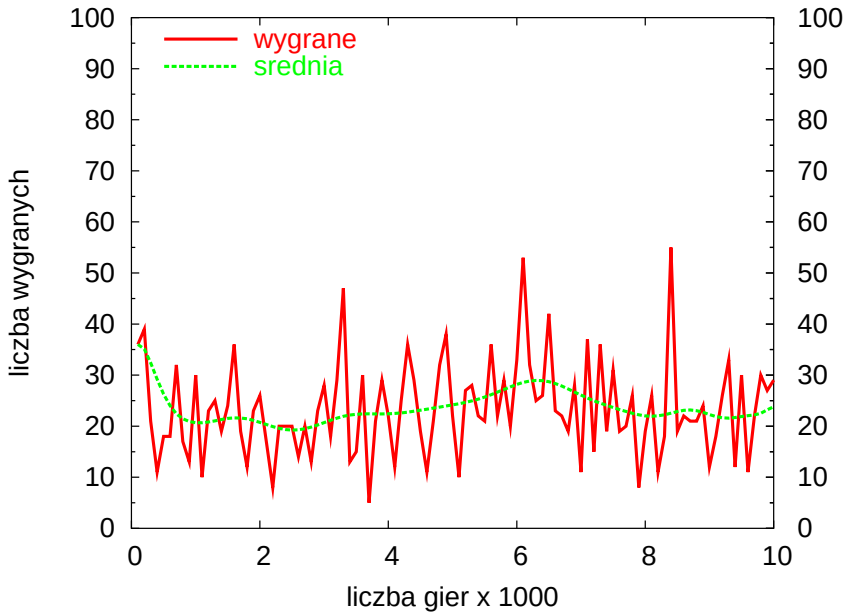
Zwiększenie głębokości przeszukiwania drzewa gry

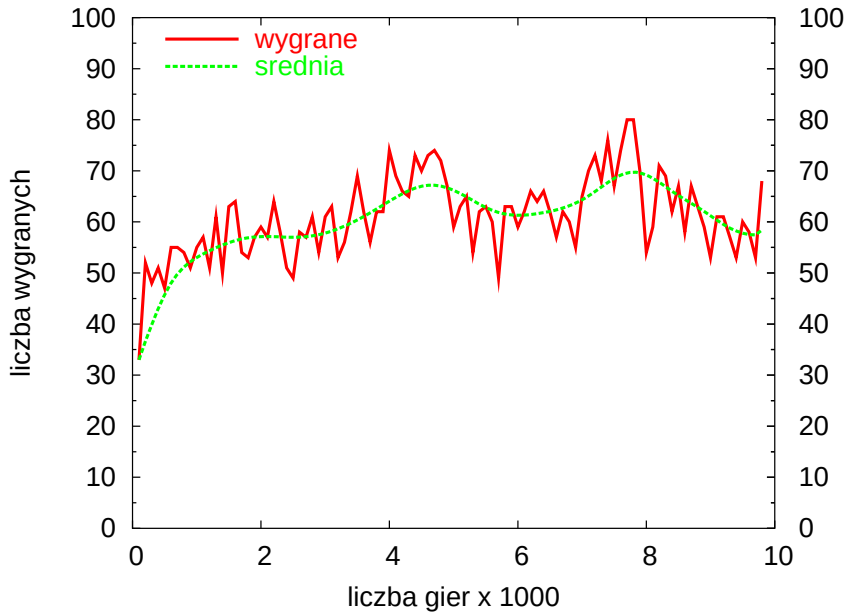




Głębokość przeszukiwania drzewa gry $d = 6$

Zanotowany postęp w nauczaniu





Gracz losowy - gracz uczący się (trening)

Porównanie z innymi programami

zestaw współczynników	wygrane	przegrane	remisy	wynik dla TD-GAC
program 1 - poziom *novice*				
TD-GAC (niski poziom)	0	10	0	0 : 10
TD-GAC (średni poziom)	0	10	0	0 : 10
TD-GAC (wysoki poziom)	1	2	7	$4\frac{1}{2} : 5\frac{1}{2}$
program 2 - poziom *weak*				
TD-GAC (niski poziom)	0	0	1	$\frac{1}{2} : \frac{1}{2}$
TD-GAC (średni poziom)	1	0	0	1 : 0
TD-GAC (wysoki poziom)	0	0	1	$\frac{1}{2} : \frac{1}{2}$
program 2 - poziom *normal*				
TD-GAC (niski poziom)	0	1	0	0 : 1
TD-GAC (średni poziom)	1	0	0	1 : 0
TD-GAC (wysoki poziom)	0	0	1	$\frac{1}{2} : \frac{1}{2}$
program 2 - poziom *strong*				
TD-GAC (niski poziom)	0	1	0	0 : 1
TD-GAC (średni poziom)	0	1	0	0 : 1
TD-GAC (wysoki poziom)	0	0	1	$\frac{1}{2} : \frac{1}{2}$

Gra z ludźmi

zestaw współczynników	wygrane	przegrane	remisy	wynik dla TD-GAC
gracz 1				
TD-GAC (niski poziom)	3	2	5	$5\frac{1}{2} : 4\frac{1}{2}$
TD-GAC (średni poziom)	3	4	3	$4\frac{1}{2} : 5\frac{1}{2}$
TD-GAC (wysoki poziom)	9	0	1	$9\frac{1}{2} : \frac{1}{2}$
gracz 2				
TD-GAC (niski poziom)	5	4	1	$5\frac{1}{2} : 4\frac{1}{2}$
TD-GAC (średni poziom)	6	4	0	6 : 4
TD-GAC (wysoki poziom)	8	2	0	8 : 2
gracz 3				
TD-GAC (niski poziom)	3	5	2	4 : 6
TD-GAC (średni poziom)	5	5	0	5 : 5
TD-GAC (wysoki poziom)	3	5	2	4 : 6
gracz 4				
TD-GAC (niski poziom)	2	6	2	3 : 7
TD-GAC (średni poziom)	0	7	3	$1\frac{1}{2} : 8\frac{1}{2}$
TD-GAC (wysoki poziom)	4	5	1	$4\frac{1}{2} : 5\frac{1}{2}$

Arthur L. Samuel

- Żył w latach 1901-1990
- Jeden z pionierów sztucznej inteligencji
- Napisał program grający w warcaby w 1959
- **Rote learning**
- **Generalization learning**
  (Temporal difference learning)

Podsumowanie

- $TD(\lambda)$ jako obiecująca metoda nauczania
- Automatyzacja doboru wag bez udziału człowieka
- Wiele miejsca na poprawę
  - wieksza głębokość przeszukiwania drzewa gry
  - różne funkcje oceny w zależności od sytuacji na planszy