

A stylized logo of a plant with a vertical stem and several large, dark blue leaves. The stem is a gradient from orange at the bottom to dark blue at the top. The leaves are layered and curved, creating a sense of movement.

MAGICIAN: GUIDED UCT

Ekstrakcja uniwersalnej funkcji
ewaluacyjnej z rezultatów symulacji
UCT

Karol Walędzik



GENERAL GAME PLAYING





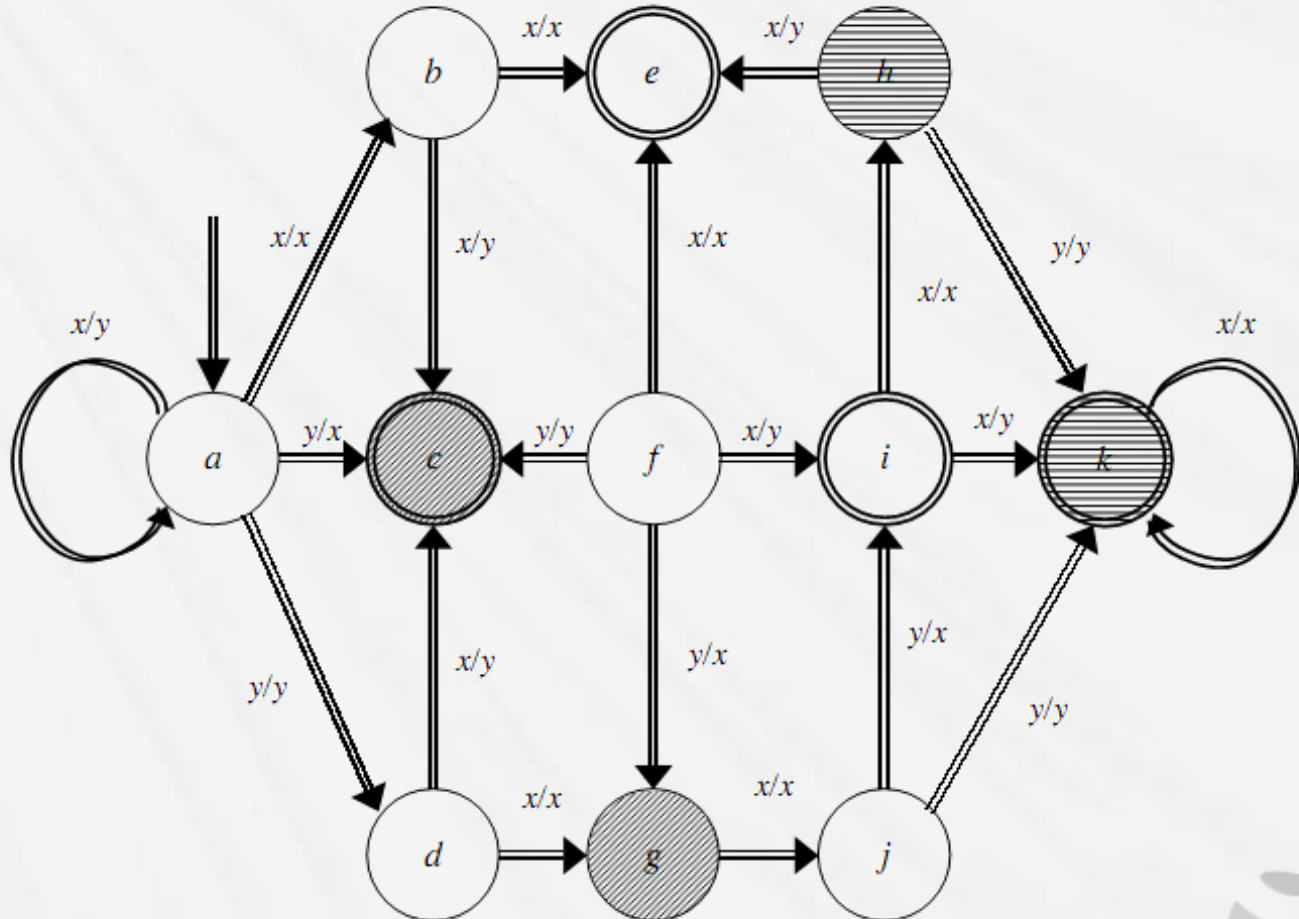
General Game Playing?

- ❖ Cel:
 - ◆ stworzenie systemu umiejącego grać/nauczyć się grać we „wszystkie” gry
- ❖ Turniej w ramach AAAI National Conference
 - ◆ corocznie od 2005 roku
- ❖ Przebieg rozgrywki w ramach turnieju:
 - ◆ prezentacja zasad i czas na ich analizę (np. 5m)
 - ◆ rozgrywka z ograniczeniem czasowym na wykonanie pojedynczego ruchu

Model gry

- ❖ Skończona
- ❖ Skończona liczba graczy
 - ◆ w tym 1
 - ◆ czyli *de facto* łamigłówka
- ❖ synchroniczna
 - ◆ wszyscy gracze ruszają się równocześnie (ale dopuszczalne ruchy typu *noop*)
- ❖ skończona liczba legalnych ruchów w każdym ze skończonej liczby stanów
- ❖ zmiany stanu tylko w wyniku ruchów
- ❖ → maszyna stanowa

Model gry c.d.



M. Genesereth, N. Love, and B. Pell. General Game Playing: Overview of the AAI Competition. AI Magazine, 26(2):62-72, 2005.



Game Definition Language (GDL)

- ❖ Opis gry jako maszyny stanowej: zbyt rozwlekły
- ❖ Krótszy sposób opisu: Game Definition Language (GDL)
 - ◆ opis gry za pomocą formuł logicznych
 - ◆ język bazujący na zmodyfikowanym Datalogu
 - ◆ Datalog to podzbiór Prologa
 - ◆ podstawowe relacje:
 - ◆ *role, true, init, next, legal, does, goal, terminal*

GDL: Tic-Tac-Toe

```
1. (role xplayer)
2. (role oplayer)
3. (init (cell 1 1 b))
4. (init (cell 1 2 b))
...
5. (init (cell 3 3 b))
6. (init (control xplayer))
7. (<= (next (cell ?m ?n x))
8.     (does xplayer (mark ?m ?n))
9.     (true (cell ?m ?n b)))
10. (<= (next (control xplayer))
11.     (true (control oplayer)))
...
13. (<= (row ?m ?x)
14.     (true (cell ?m 1 ?x))
15.     (true (cell ?m 2 ?x))
16.     (true (cell ?m 3 ?x)))
17. (<= (line ?x)
18.     (row ?m ?x))
...
19. (<= (legal ?w (mark ?x ?y))
20.     (true (cell ?x ?y b))
21.     (true (control ?w)))
22. (<= (legal xplayer noop)
23.     (true (control oplayer)))
...
24. (<= (goal xplayer 0) (line o))
25. (<= (goal oplayer 100) (line o))
...
26. (<= terminal (line x))
```

J. Reisinger, E. Bahceci, I. Karpov, and R. Miikkulainen. Coevolving strategies for general game playing. In Proceedings of the IEEE Symposium on Computational Intelligence and Games (CIG 2007), 320-327, Honolulu, Hawaii, 2007. IEEE Press.

Multi-
Approach
Game
Independent
Cunningly
Intelligent
Arch-
Nemesis





**UCT
GUIDED UCT**

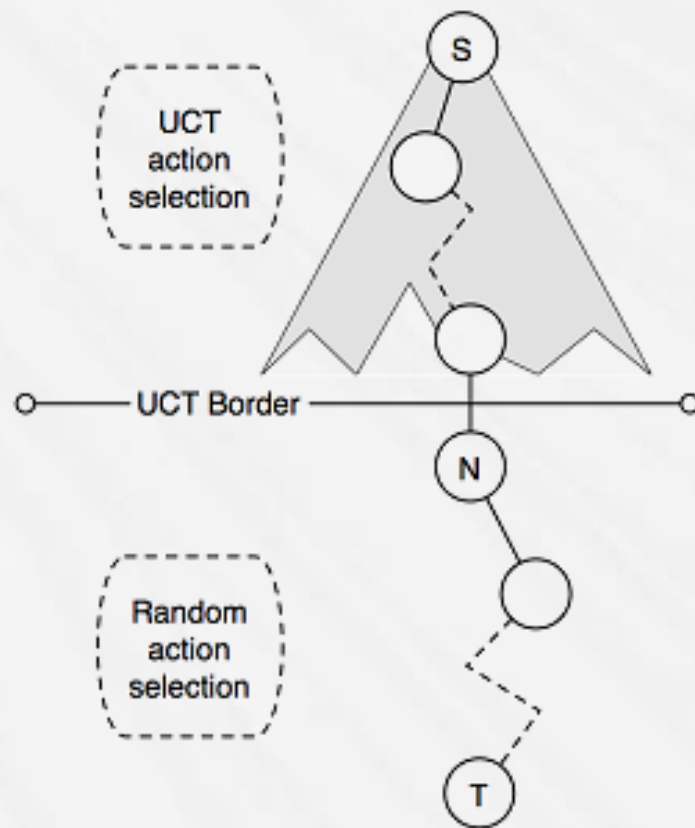
UCT

- ❖ Sposób wybierania kolejnego ruchu podczas symulacji (UCT):

$$a^* = \operatorname{argmax}_{a \in A(s)} \left\{ Q(s, a) + C \sqrt{\frac{\ln N(s)}{N(s, a)}} \right\}$$

- ◆ $Q(s, a)$ – średni dotychczasowy wynik pary stan-ruch
- ◆ N – liczba wizyt w danym stanie/wykonań danej akcji
- ◆ akcje nigdy nie wykonane wybierane są w pierwszej kolejności

UCT c.d.



UCT c.d.

- ❖ Ograniczanie zużycia pamięci:
 - ◆ usuwanie danych o węzłach powyżej aktualnego po każdym (realnie) wykonanym ruchu
 - ◆ co z pozycjami powtórzonymi w drzewie gry
 1. duplikowanie
 2. **inteligentne wykrywanie, kiedy można je usunąć z pamięci**
 - ◆ dodawanie tylko jednego nowego węzła per symulacja
- ❖ Modelowanie przeciwnika
 - ◆ każdy przeciwnik ma przydzielony swój własny model (*Cadia*)
 - ◆ przeciwnicy podejmują losowe decyzje



UCT - usprawnienia

- ❖ CADIAPlayer:

- ◆ *history heuristic*

- ◆ modyfikuje prawdopodobieństwo wybrania niewypróbowanej akcji (zarówno w fazie UCT, jak i MC)

Guided UCT

- ❖ Guided UCT + funkcja ewaluacyjna
- ❖ Gdzie ta funkcja ewaluacyjna?
 1. W fazie MC:
 - a) całkowite zastąpienia fazy
 - ◆ (z zadaniem prawdopodobieństwem)
 - b) **zakończenie symulacji przed osiągnięciem końca gry**
 - ◆ z zadaniem prawdopodobieństwem w każdym węźle drzewa

Guided UCT

- ❖ Guided UCT + funkcja ewaluacyjna
- ❖ Gdzie ta funkcja ewaluacyjna?

2. W fazie UCT:

- a) jako wstępne sortowanie ruchów nigdy nie wypróbowanych
- b) jako startowy szacunek wartości ruchów z wagą odpowiadającą n symulacjom

◆ Problem:

- ◆ porównywanie bardzo podobnych pozycji (różniących się tylko ostatnim ruchem)



**STATISTICAL GUCT:
HEURYSTYCZNA FUNKCJA EWALUACYJNA**

Generowanie komponentów

- ❖ Lista podstawowa:
 - ◆ wyrażenia występujące w opisie gry
 - ◆ cechy nie muszą być binarne
 - ◆ zliczanie sposobów rozwiązania zmiennych w wyrażeniu
- ❖ Uzupełnianie listy:
 - ❖ uogólniane przez podstawianie zmiennych za stałe i możliwych stałych za zmienne

`cell (1, 1, x)`

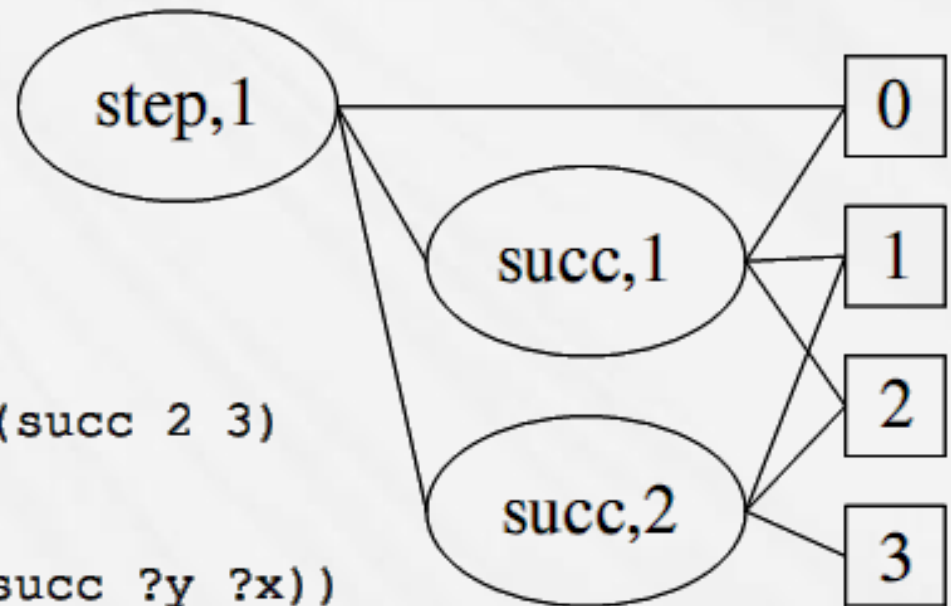
`cell (1, 1, ?)`

`cell (?, ?, x)`

`cell (1, ?, x)`

Generowanie komponentów c.d.

- ❖ „Podstawianie stałych za zmienne”
 - ◆ Wszystkich??
 - ◆ Wyznaczanie dziedzin argumentów predykatów/funkcji:



```
(succ 0 1) (succ 1 2) (succ 2 3)
(init (step 0))
(<= (next (step ?x))
    (true (step ?y)) (succ ?y ?x))
```

Złożone komponenty

- ❖ Relacje między wartościami komponentów różniących się na jednej pozycji:
 - ◆ $\text{cell}(?, ?, w) - \text{cell}(?, ?, b)$
 - ◆ $\text{cell}(?, ?, wk) / \text{cell}(?, ?, bk)$
 - ◆ a jako efekt uboczny także:
 - ◆ $\text{cell}(a, 1, ?) / \text{cell}(a, 3, ?)$
- ❖ Problem: co z dzieleniem przez 0?
 - ◆ Odp. 1: 0
 - ◆ Odp. 2: podzielmy przez $\frac{1}{2}$ zamiast tego
 - ◆ **Odp. 3: zrezygnujemy na razie z proporcji**




Analiza statystyczna komponentów

- ❖ Puła przykładowych sekwencji
 - ◆ ok. 100
 - ◆ początek sekwencji: wynik losowej gry
 - ◆ długość: do 5 pozycji
 - ◆ odległość między pozycjami: 1-2 ruchy
 - ◆ generowanie sekwencji: losowe ruchy
 - ◆ powiązanie sekwencji ze średnim wynikiem: symulacje Monte Carlo (~10)



Analiza statystyczna komponentów c.d.

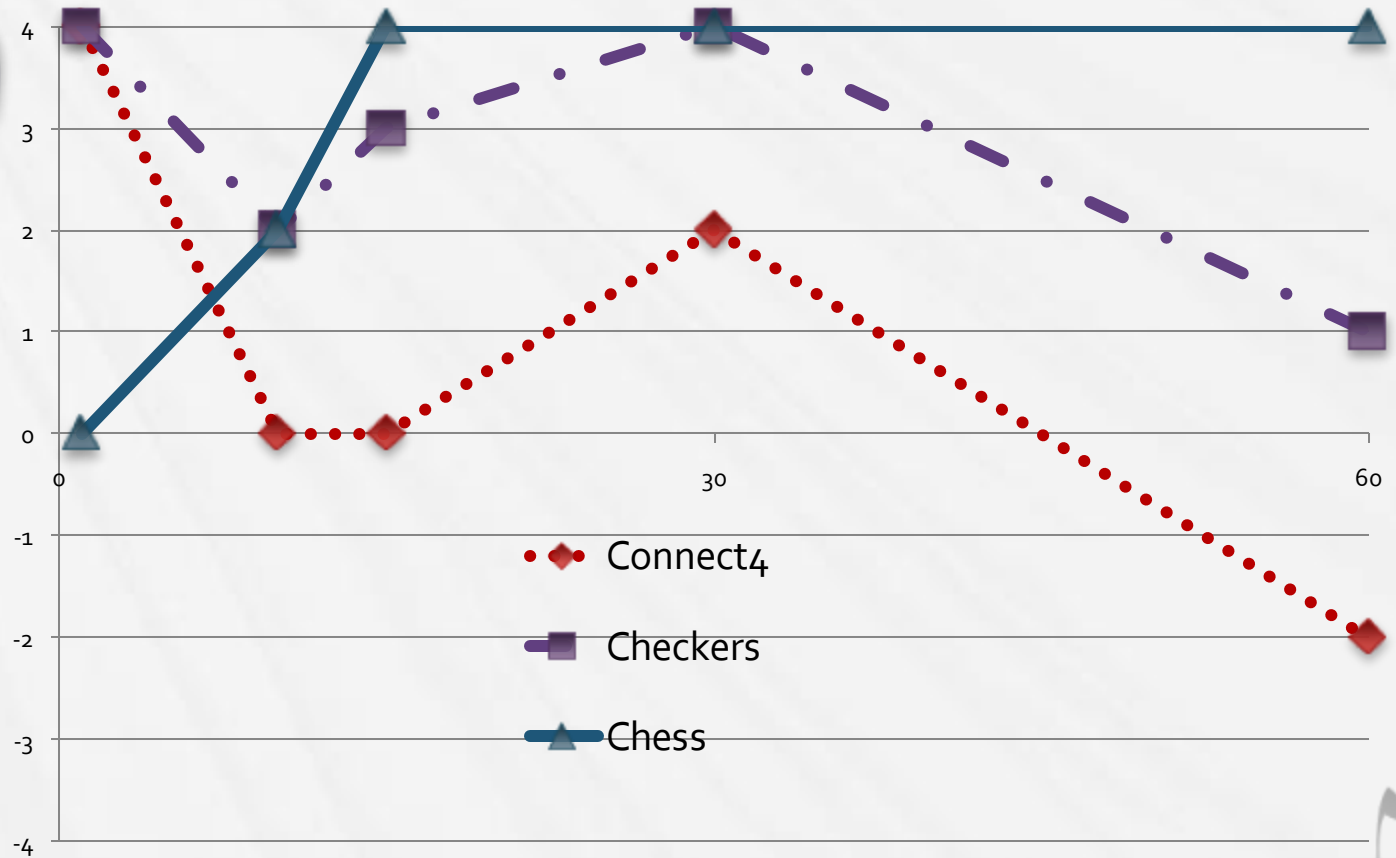
- ❖ Średnie wartości i wariancje wartości komponentów
 - ◆ dla całej gry
 - ◆ uwzględnia tylko pierwszą pozycję z każdej sekwencji
 - ◆ dla poszczególnych sekwencji
- ❖ Korelacja wartości komponentów z wynikiem gry
 - ◆ brana pod uwagę średnia wartość komponentu w sekwencji
- ❖ Stabilność wartości komponentów
 - ◆ iloraz wariancji między początkami sekwencji oraz średniej wariancji w ramach sekwencji
 - ◆ $S = TV / (TV + 10 * SV)$



Funkcja ewaluacyjna - heurystyczna

- ❖ Wybór komponentów:
 - ◆ sortowanie:
 - ◆ $\min(\text{stabilność}, |\text{korelacja z wynikiem}|)$
 - ◆ obcięcie do 30 pierwszych komponentów
- ❖ Wagi:
 - ◆ $\text{stabilność} * \text{korelacja z wynikiem}$
- ❖ Składnik stały:
 - ◆ średni wynik

StatGUCT vs. UCT



Przykładowa funkcja ewaluacyjna (warcaby)

$$\begin{aligned} & \diamond -0,087 * \text{cell}(a, 4, \text{wp})\{\text{int}\} + -0,1 * \text{cell}(a, ?, \text{wp})\{\text{int}\} + \\ & 0,12 * \text{cell}(h, 3, \text{bp})\{\text{int}\} + 0,056 * \text{cell}(f, 3, \text{bp})\{\text{int}\} + \\ & -0,045 * \text{cell}(g, 8, \text{wk})\{\text{int}\} + 0,052 * \text{cell}(f, 1, \text{bk})\{\text{int}\} + \\ & -0,051 * \text{cell}(?, 8, \text{wk})\{\text{int}\} + -0,041 * \text{cell}(g, ?, \text{wk})\{\text{int}\} + \\ & 0,036 * \text{cell}(a, 2, \text{bk})\{\text{int}\} + \mathbf{0,13 * \text{cell}(?, ?, \text{bk})\{\text{int}\}} + \\ & 0,046 * \text{cell}(?, 1, \text{bk})\{\text{int}\} + \mathbf{-0,098 * \text{cell}(?, ?, \text{wk})\{\text{int}\}} + \\ & 0,039 * \text{cell}(a, ?, \text{bk})\{\text{int}\} + 0,11 * \text{cell}(f, 5, \text{wk})\{\text{int}\} + \\ & 0,036 * \text{cell}(d, ?, \text{bk})\{\text{int}\} + -0,064 * \text{cell}(a, 2, \text{wp})\{\text{int}\} + \\ & 0,056 * \text{cell}(?, 3, \text{bp})\{\text{int}\} + -0,032 * \text{cell}(g, 6, \text{wp})\{\text{int}\} + \\ & 0,026 * \text{cell}(?, 6, \text{bk})\{\text{int}\} + -0,051 * \text{cell}(?, 5, \text{wp})\{\text{int}\} + \\ & -0,037 * \text{cell}(?, 6, \text{wp})\{\text{int}\} + 0,038 * \text{cell}(d, 3, \text{bk})\{\text{int}\} + \\ & \mathbf{-0,12 * \text{cell}(?, ?, \text{wp})\{\text{int}\}} + 0,055 * \text{cell}(g, ?, \text{bp})\{\text{int}\} + \\ & -0,02 * \text{cell}(?, 7, \text{wk})\{\text{int}\} + 0,027 * \text{cell}(d, 7, b)\{\text{int}\} + \\ & -0,044 * \text{cell}(h, 3, \text{bk})\{\text{int}\} + 0,033 * \text{cell}(f, ?, \text{bk})\{\text{int}\} + \\ & 0,037 * \text{cell}(b, 3, \text{bk})\{\text{int}\} + -0,074 * \text{cell}(h, 1, \text{wp})\{\text{int}\} + 50 * 1 \end{aligned}$$



**SELF-GUIDED UCT:
NAUKA**



Self-Guided UCT

- ❖ UCT uczy się o konkretnych ruchach w konkretnych pozycjach
- ❖ Pomysł: uogólnijmy tę wiedzę do ogólnej funkcji ewaluacyjnej
- ❖ Nauka:
 - ◆ offline – w fazie analizy gry
 - ◆ online – w czasie ruchu przeciwnika



SGUCT - wersja alfa

- ❖ Zbiór uczący:
 - ◆ stany, w których przynajmniej raz wypróbowano wszystkie ruchy
 - ◆ wartość pozycji = średnia wartość najlepszego ruchu
- ❖ Populacja uczona:
 - ◆ 10 osobników – funkcji ewaluacyjnych
 - ◆ stworzone na podstawie analizy statystycznej



SGUCT - wersja alfa

- ❖ Uczenie gradientowe, podzielone na pokolenia
 - ◆ pokolenie = 100 epok
 - ◆ przystosowanie = odwrotność MSE na zbiorze testowym
 - ◆ selekcja:
 - ◆ usuń 3 najłabsze osobniki
 - ◆ sklonuj 3 najlepsze z mutacją
 - ◆ potencjalnie zmutuj wszystkie osobniki oprócz 3 najlepszych



SGUCT - wersja alfa

- ❖ Mutacja

- ◆ mutacja wag ($p = 0.3$)
 - ◆ każda waga mutowana z $p = 0.1$
 - ◆ mutacja: dodanie losowej wartości z przedziału $[-\delta; \delta]$
 $\delta = \max(1, |\text{aktualna waga}|)$

SGUCT - wersja alfa

❖ Mutacja

◆ mutacja cech ($p = 0.2$)

◆ wrażliwość = waga * max(wartość cechy)

◆ potencjalnie usuń cechę:

◆ warunek:

◆ osobnik zbyt duży lub

◆ $\max(\text{wrażliwości}) / \min(\text{wrażliwości}) > 100$ lub

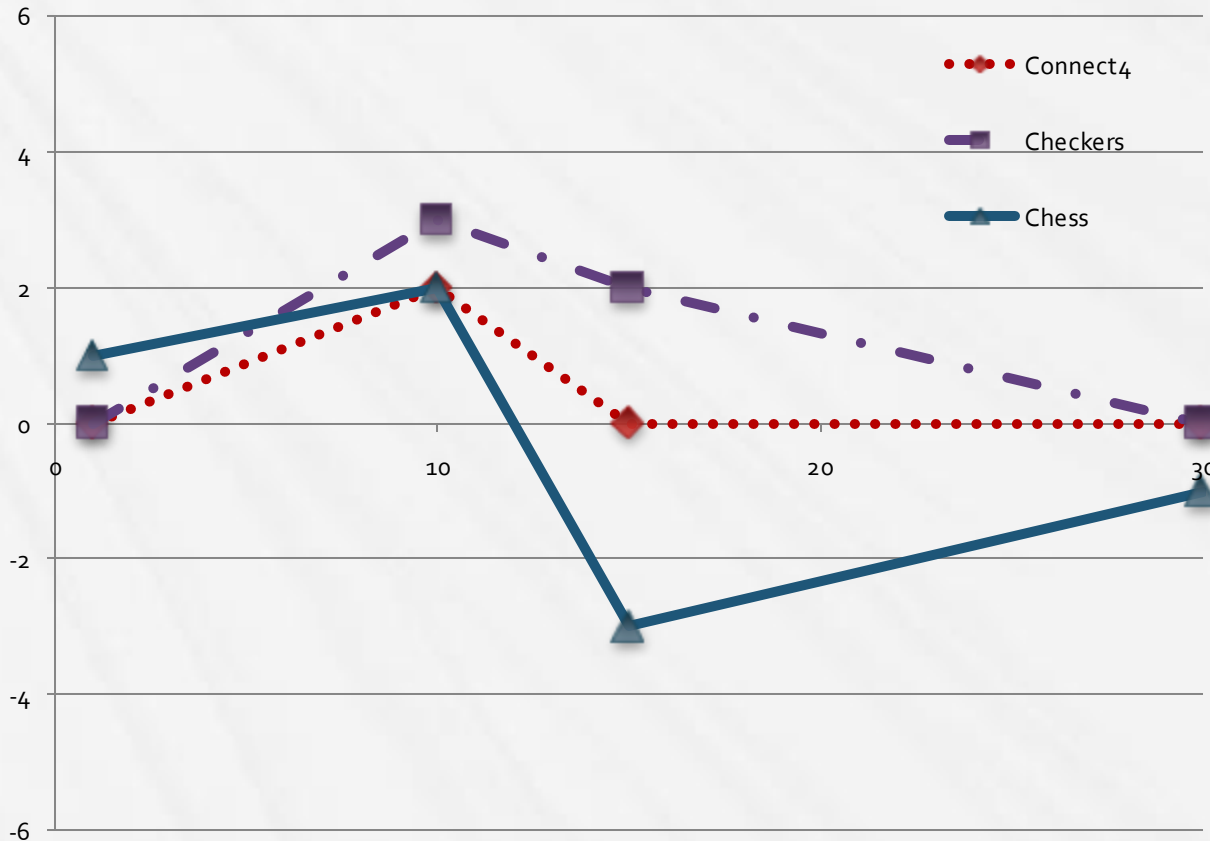
◆ zaszło zdarzenie o $p = 0.6$

◆ usuwana cecha: wybór ruletkowy na podstawie odwrotności wrażliwości

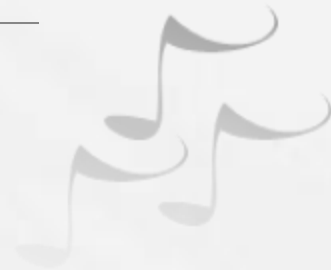
◆ przypisz nową cechę:

◆ wygeneruj ją jak przy tworzeniu nowego osobnika

SGUCT vs. StatGUCT



UWAGA: Mała powtarzalność wyników



Bibliografia

- ❖ J. Clune: Heuristic evaluation functions for General Game Playing. In Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence (AAAI-07), pages 1134–1139, Vancouver, BC, Canada, 2007. AAAI Press.
- ❖ H. Finnsson, Y. Björnsson: Simulation-based approach to General Game Playing. In Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence (AAAI-08), pages 259–264, Chicago, IL, 2008. AAAI Press.
- ❖ General Game Playing website by Stanford University. <http://games.stanford.edu/>.
- ❖ General Game Playing website by Dresden University of Technology. <http://www.general-game-playing.de/>.
- ❖ M. Genesereth, N. Love: General Game Playing: Overview of the AAAI Competition. <http://games.stanford.edu/competition/misc/aaai.pdf>, 2005.
- ❖ D. Kaiser: The Design and Implementation of a Successful General Game Playing Agent. In Proceedings of FLAIRS Conference, pages 110–115, 2007.
- ❖ G. Kuhlmann, K. Dresner, and P. Stone: Automatic heuristic construction in a complete General Game Player. In Proceedings of the Twenty-First AAAI Conference on Artificial Intelligence (AAAI-06), pages 1457–1462, Boston, MA, 2006. AAAI Press.
- ❖ N. Love, T. Hinrichs, D. Haley, E. Schkufza, M. Genesereth: General Game Playing: Game Description Language Specification. <http://games.stanford.edu/language/spec/gdl-spec-2008-03.pdf>, 2008.
- ❖ J. Mańdziuk, M. Kusiak, K. Walędzik: Evolutionary-based heuristic generators for checkers and give-away checkers. *Expert Systems*, 24(4): 189–211, Blackwell-Publishing, 2007.
- ❖ J. Reisinger, E. Bahceci, I. Karpov and R. Miikkulainen: Coevolving strategies for general game playing. In Proceedings of the IEEE Symposium on Computational Intelligence and Games (CIG'07), pages 320–327, Honolulu, Hawaii, 2007. IEEE Press.
- ❖ S. Schiffel, M. Thielscher: Automatic Construction of a Heuristic Search Function for General Game Playing. In Seventh IJCAI International Workshop on Nonmonotonic Reasoning, Action and Change (NRAC07).