Porównanie rozwiązań równowagowych Stackelberga w grach z wynikami stosowania algorytmu UCT

Jan Karwowski

Zakład Sztucznej Inteligencji i Metod Obliczeniowych
Wydział Matematyki i Nauk Informacyjnych PW

24 VI 2015

Security Games

Zastosowania

- Zabezpieczenie lotnisk przed zamachami
- Zabezpieczenie ruchomych celów
- Zabezpieczenie zasobów przed kradzieżą

Model miejsca rozgrywki

- Zbiór miejsc, bez zadanych relacji przestrzennych
- **Graf**
- Przestrzeń ciągła

Gracze

- Broniący
- Atakujący (jeden lub wielu)

Pursuer-Evader Game

Gracze

- Ścigany
- Ścigający — cel: znalezienie się w tym samym miejscu i czasie co Ścigany
- Grają przeciw sobie

- Czas (długość) gry może nie być ściśle określony
- Może się rozgrywać na grafie (znalezienie się w tym samym wierzchołku) lub w przestrzeni ciągłej (znalezienie się bliżej niż ϵ w pewnej metryce)

Model I

Wybrany arbitralnie.

Zarys

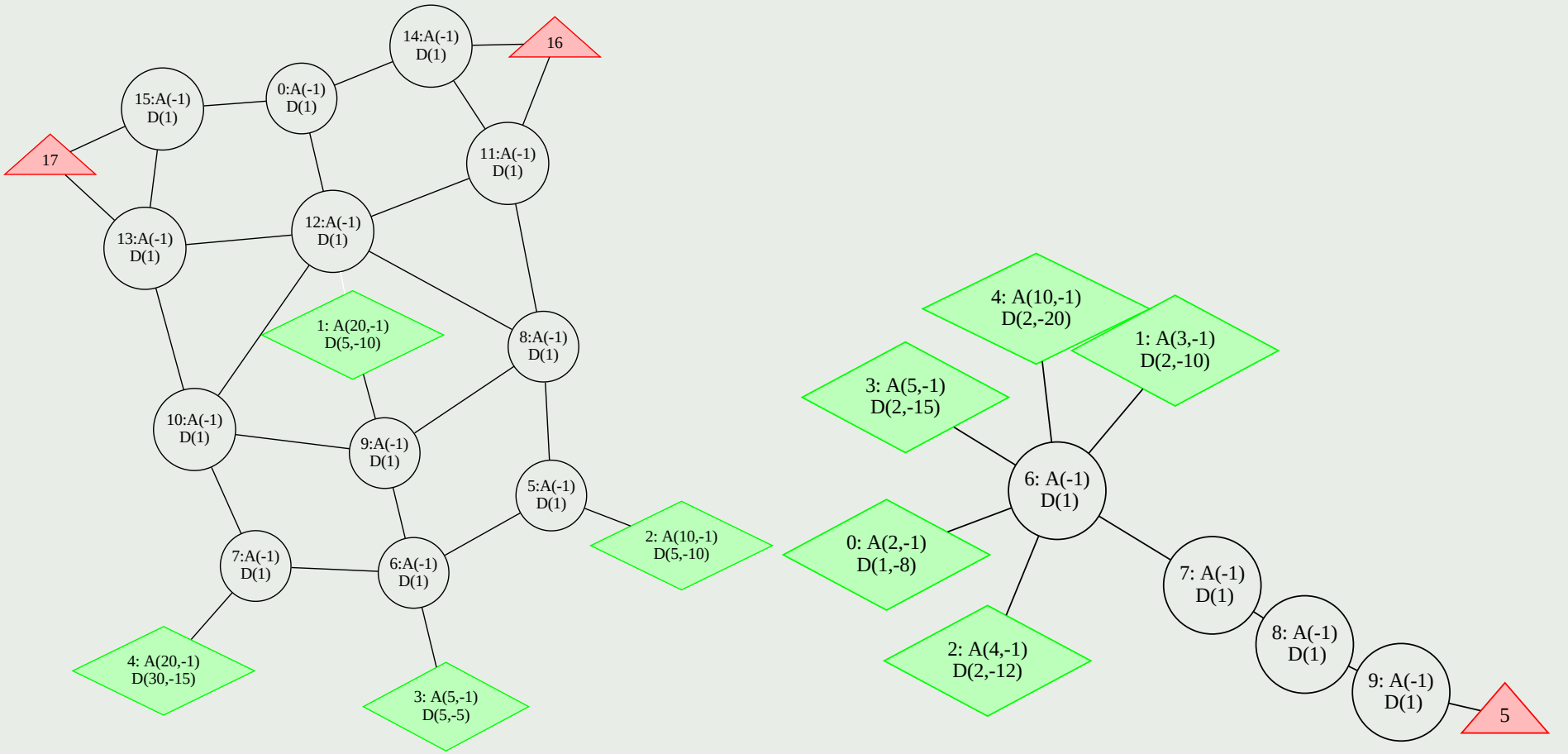
- Graf skierowany
- Wierzchołki: zwykłe, cele, punkty startowe, baza
- Krawędzie – możliwości przemieszczenia
- Brak limitu czasu

Plansza

- $G = (V, E)$, $E \subseteq \{(v, u) | v, u \in V, v \neq u\}$
- $T \subset V$ – cele, $S = subsetV$ – punkty startowe, $b \in V$ – baza obrońców, $(T \cup \{b\}) \cap S = \emptyset$.
- $R_A^+(t)$, $R_A^-(v)$, $R_D^+(v)$, $R_D^-(t)$ – wypłaty w wierzchołkach/celach

Model II

Przykłady planszy, 0 – baza



Model III

Obrońca

- Stała liczba
- W każdej kolejce przesuwa każdą jednostkę na sąsiedni wierzchołek lub pozostaje w miejscu

Atakujący

- W każdej kolejce może wprowadzić na planszę nową jednostkę.
- W każdej kolejce przesuwa każdą jednostkę na sąsiedni wierzchołek lub pozostaje w miejscu
- Jednostki są usuwane po złapaniu lub dotarciu do celu

Model IV

Kolejka (Kolejność operacji ma znaczenie)

1. Każdy z graczy, niezależnie od siebie wybiera nowe pozycje dla swoich jednostek
2. Następuje przesunięcie
3. Jednostki ataku, które są w tym samym wierzchołku co jakaś jednostka obrony są zdjęte z planszy. Wyliczane są odpowiednie wypłaty.
4. Jednostki ataku, które są w celu przeprowadziły skuteczny atak. Wyliczane są odpowiednie wypłaty, jednostki są usuwane z planszy.

Implementacja modelu (widok obrońcy)

Wariant B

Stan Wektor od długości równej liczbie obrońców, wartości – bieżący wierzchołek

Ruch Wektor tej samej długości – nowe pozycje

Opcjonalnie: historia incydentów

UCT – budowa drzewa na starcie gry

Nigdy nie kopiujemy bieżącego stanu w symulacjach. Dochodzimy do n-tego poziomu drzewa.

Algorithm: Przebieg modyfikacji UCT dla gier bez pełnej informacji

```
for depth ← 1 · · · maxDepth do
    for i ← 1 . . . simCount do
        // Wykonuje odpowiednio dużą liczbę symulacji na
           poziomie depth
        State ← InitialState
        for d ← 1 · · · depth − 1 do
            // Zejście do poziomu depth w drzewie
            Best ← UCTBestMove(State)
            MakeMove(State, Best)
        SingleRun(State)
```

Całość powtarzamy n razy.

Atakujący (przykładowy)

Przewidywanie znalezienia się obrońcy w celu

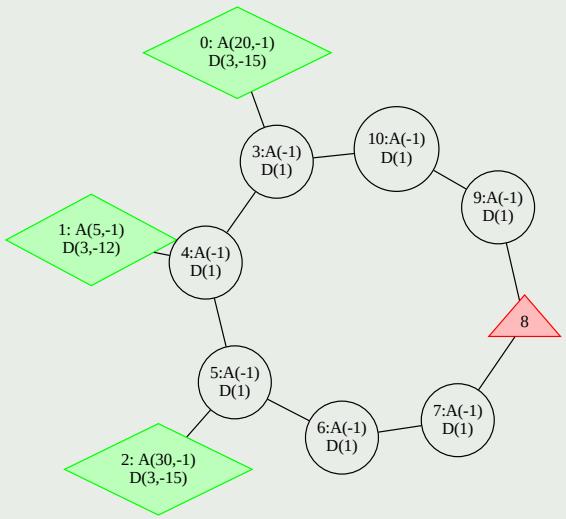
- Rejestruje fakt pojawienia się atakującego w celu
- Dla każdego celu oblicza $E(j) = (R_j(t - d_j) + P_j d_j)/t$, t – długość obserwacji, R_j – nagroda, P_j – kara, d_j – liczba obserwacji.
- Z prawdopodobieństwem $0,2$ wyprowadza atak z losowego startu do losowego celu. Start – rozkład równomierny, cel – ruletka z prawd. $E(j)$. Wpuszczony atak nie zmienia celu.

Dodatkowe możliwości

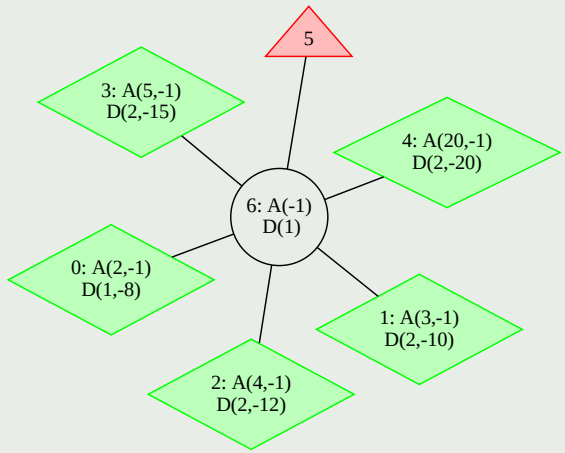
- Atakujący wie gdzie jest baza obrońców
- Atakujący odróżnia jednostki obrońcy
- Atakujący widzi obrońców w celach
- Atakujący widzi obrońców, którzy właśnie go złapali
- Atakujący zna wypłaty w celach

Gry testowe

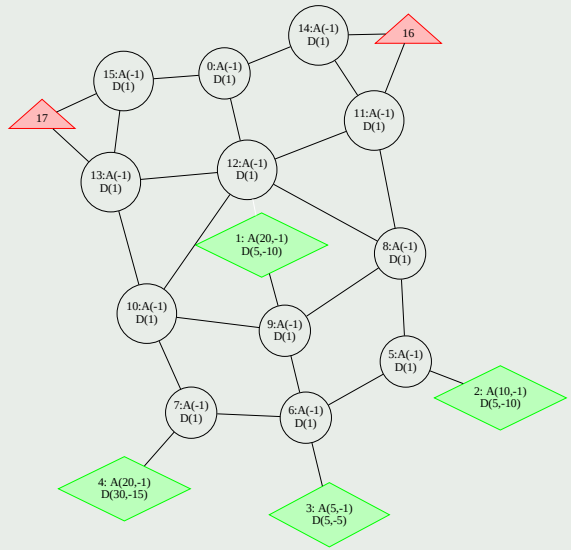
Gra 3: 2 obrońców



Gra 4: 1 oobrońca



Gra 5: 2 obrońców



Liczba skutecznych ataków

Game	Config set	Attack Situations			Overall pay-off			Random
		mean	med	stddev	mean	med	stddev	avg pay-off
3A	ex20-len2	0	0	0	102455.8	101805	2218.7	-547124
3A	ex15-len2	1	0	2.9	104847.4	101552	6519.9	-547124
3A	ex12-len4	0.8	0	1.5	107871.6	103169	12187	-547124
3A	ex7-len2	908.7	0	2863.4	82902.6	80123	48089.9	-547124
3A	ex4-len3	2486.2	5	4569	52859.1	79923	62337.1	-547124
3B	ex15-len2	0	0	0	102553.3	100966	3197.3	-544089
3B	ex12-len4	0.2	0	1.1	106392.3	104069	10478.7	-544089
3B	ex7-len2	557.7	4	2536.1	107031.5	103290	22436.9	-544089
3B	ex4-len3	1052.5	4	3311.9	75123	80126	45902.9	-544089
4A	ex12-len4	0	0	0	79981.9	79987	241.7	-674773
4A	ex7-len2	1907.2	0	8739.8	51276	79944	131656.6	-674773
4A	ex4-len3	3806.5	0	12022.3	19356	80125	192247.8	-674773
4B	ex7-len2	0	0	0	79950.3	79942	239.5	-674925.6
4B	ex4-len3	3829.9	0	12096	19511.1	79862	190970.3	-674925.6
5A	ex20-len2	0.8	0	1.9	80010.1	80064	218.9	-522846.5
5A	ex15-len2	1.2	0	2.1	80062	80118	274.5	-522846.5
5A	ex12-len4	356.6	4	1618.9	76033	79885	17669.8	-522846.5
5A	ex7-len2	1928	0	5852.2	59649.7	79951	60579.8	-522846.5
5A	ex4-len3	9084.2	12	11624.6	-20358.3	79538	125375	-522846.5
5B	ex20-len2	2089.6	4	6617.7	83586.1	79908	13466	-522190.5
5B	ex15-len2	2	0	3.1	79936.9	79948	194	-522190.5
5B	ex12-len4	1169	2	5341.7	81385.2	79980	6734.9	-522190.5
5B	ex7-len2	1041.3	6	2596.7	68444.9	79836	28531.5	-522190.5
5B	ex4-len3	8270.5	21	10791.5	-10813	79423	118137.9	-522190.5

Dobór współczynników

Najlepszy ex dla danego len

	bA	l2A	l3A	l4A	l5A	e2A	e3A	e4A	e5A	bB	l2B	l3B	l4B	l5B	e2B	e3B	e4B	e5B
2	10	10	10	12	12	7	10	10	7	4	4	7	10	10	4	7	10	7
3	10	10	10	10	15	8	12	8	15	4	4	5	7	10	5	8	4	7
5	10	10	10	10	10	10	10	8	15	5	5	7	8	8	5	8	5	12
10	10	10	10	10	10	10	8	7	12	4	5	8	7	7	4	8	7	12
20	10	10	10	10	10	10	12	12	12	4	8	7	8	12	4	8	12	7
40	10	10	12	12	10	10	12	15	15	4	5	7	10	12	4	5	15	8

Najlepszy len dla danego ex

	bA	l2A	l3A	l4A	l5A	e2A	e3A	e4A	e5A	bB	l2B	l3B	l4B	l5B	e2B	e3B	e4B	e5B
4	10	10	10	20	2	40	10	2	20	10	20	2	10	20	10	5	10	20
5	10	10	10	2	20	40	2	10	20	5	10	2	10	20	3	40	5	20
7	10	10	10	10	10	2	20	10	2	5	5	2	3	10	5	2	10	2
8	10	2	10	10	2	10	10	3	3	5	20	10	5	5	5	10	40	40
10	10	10	10	10	10	10	2	2	40	10	20	2	40	3	40	10	5	2
12	10	10	10	10	10	10	20	20	20	3	5	10	40	20	40	20	20	20
15	2	10	10	10	10	20	2	40	5	2	3	40	5	20	20	3	40	2

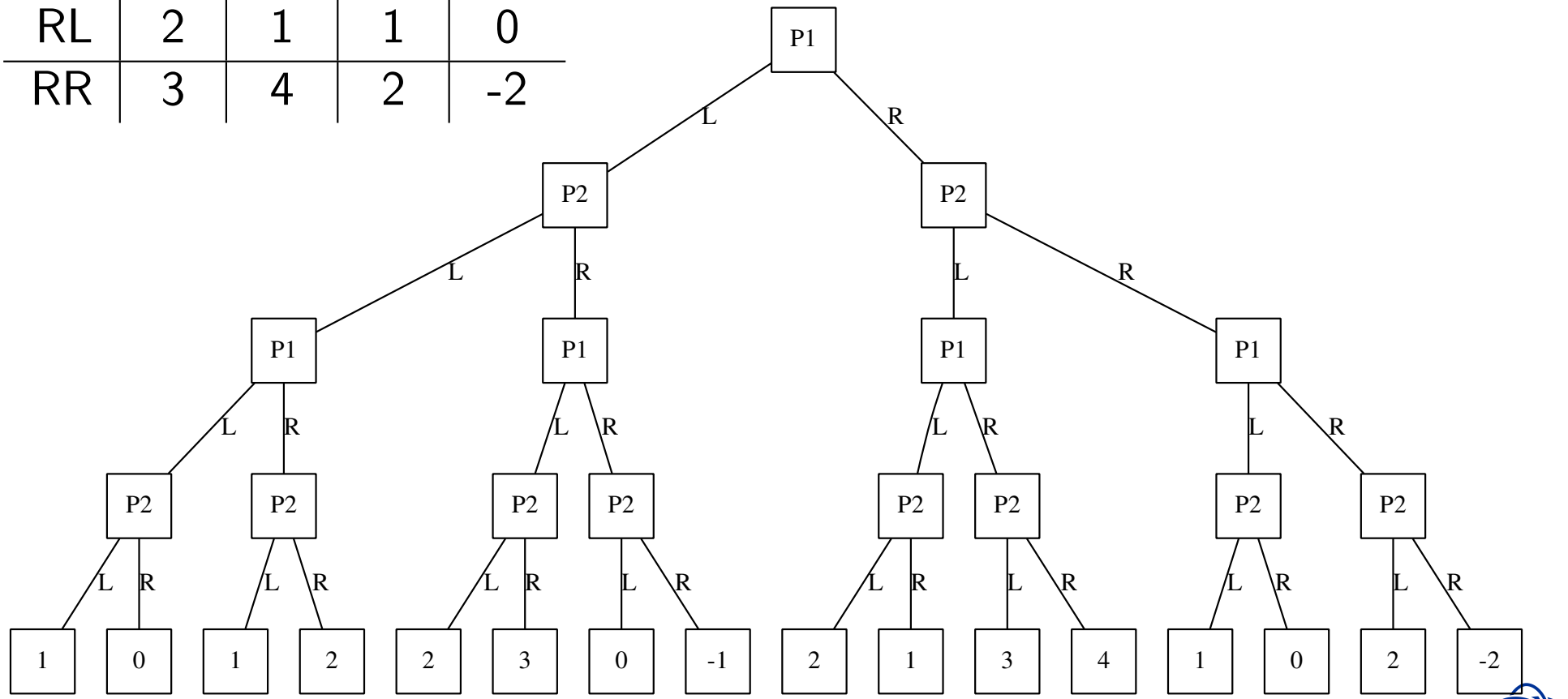
Gra w postaci normalnej (macierzowa) **Nowe**

Ruch gracza 1 \ Ruch gracza 2	M_1	M_2	M_3	M_4
m_1	5	3	2	-8
m_2	2	4	8	1
m_3	-3	-4	-2	100

- Gra o sumie zerowej (stałej) – $\pm$ jedna macierz
- Gra nie o sumie zerowej – dwie macierze

Gra wielokrokowa w postaci macierzowej **Nowe**

	LL	LR	RL	RR
LL	1	0	2	3
LR	2	3	0	-1
RL	2	1	1	0
RR	3	4	2	-2



Model Stackelberga **Nowe**

Jednokrokowa gra nie o sumie zerowej. Dwóch graczy:

- Leader (tu: obrońca)
- Follower (atakujący)

Założenie: Atakujący zna strategię obrońcy

Równowaga Stackelberga

Rozważam tylko strategie mieszane.

- Maksymalizujemy wypłatę obrońcy przy założeniu optymalnej odpowiedzi atakującego.
- Atakujący przy danej strategii broniącego wybiera strategię dającą najlepszą wypłatę (według swojej macierzy wypłat. W przypadku równoważnych wypłat wybiera opcję *korzystniejszą* dla obrońcy.
- Spostrzeżenie: zawsze istnieje deterministyczna strategia atakującego przy ustalonej strategii obrońcy (jeden najlepszy ruch).

Basar and Olsder 1982

Model Stackelberga **Nowe**

Strategia atakującego

- Policz wartości oczekiwane wszystkich ruchów (wypłaty atakującego).
- Wybierz wszystkie optymalne zagrania, policz dla nich wypłaty obrońcy.
- Wybierz dowolny z ruchów najkorzystniejszych dla obrońcy. Jako strategię zwróć ten ruch z prawdopodobieństwem 1.

Rozwiązanie **Nowe**

$$max_{q,z,a} \quad \sum_{i \in X} \sum_{j \in Q} R_{ij} z_{ij}$$

s.t.
$$\sum_{i \in X} \sum_{j \in Q} z_{ij} = 1$$
$$(\forall_{i \in X}) \sum_{j \in Q} z_{i,j} \leqslant 1$$
$$(\forall_{j \in Q}) q_j \leqslant \sum_{i \in X} z_{ij} \leqslant 1$$
$$\sum_{j \in Q} q_j = 1$$
$$(\forall_{j \in Q}) 0 \leqslant (a - \sum_{i \in X} C_{ij} (\sum_{h \in Q} z_{ih})) \leqslant (1 - q_j) M$$

$$z_{ij} \in [0, 1]$$
$$q_j \in 0, 1$$
$$a \in \mathbb{R}$$

Q – zbiór ruchów atakującego, X – zb. ruchów obrońcy, R – macierz wypłat obrońcy, C – macierz wypłat atakującego.

Paruchuri et al. 2008

Uproszczony model SG **Nowe**

Do porównania algorytmów potrzeba gry możliwej do wyrażenia w obu modelach.

- Jeden obrońca, jeden atakujący
- Gra do pierwszego incydentu lub do małego limitu kroków
- Atakujący od początku jest na planszy
- Równa liczba ruchów w każdej grze (ruchy po incydencie nie mają wpływu na wypłatę
- Open-Loop

Solver MILP **Nowe**

- `http://scip.zib.de/`
- Rozwiązanie dokładne
- Czas dla przykładowej gry nr 3 z 5 krokami: $17394s$ (prawie 5 godzin), prawie 6GB pamięci.
- Dla gry 4 – macierz problemu za duża, aby uruchomić solver dla sensownej liczby kroków (> 3).

Rozwiązanie **Nowe**

Ruch	Prawdopodobieństwo	Wypłata[1]
0, 0, 0, 0, 0	0.403846	-15
3, 4, 5, 2, 2	0.596154	3

$$E(payoff) = -4.269230769230773$$

Problem

Słaba wypłata nie oznacza, że nie należy wykonywać ruchu!!

[1] Przy optymalnej strategii atakującego

Modyfikacja UCT **Nowe** I

Atakujący

Użycie atakującego zgodnego z grą Stackelberga – konieczność wyznaczenia rozkładu prawdopodobieństwa ruchów obrońcy.

Problem

Jak zamienić statystyki UCT na rozkład prawdopodobieństwa?

Algorithm: Budowanie strategii w grze Stackelberga z użyciem UCT

$tree \leftarrow TrainUCT(randomAttacker, nil)$
for $i \leftarrow 1 \cdots m$ **do**
    $attacker \leftarrow StackelbergAttacker(GetProbabilities(tree))$
    $tree \leftarrow Damping(tree)$
    $tree \leftarrow TrainUCT(attacker, tree)$
$strategy \leftarrow GetProbabilities(tree)$

Zamiana drzewa w strategię **Nowe**

Koduję krok gry jako element stanu.

Algorithm: Prawdopodobieństwa ruchów

$movePr \leftarrow \emptyset$ **for** $leaf \in Leaves(uctTree)$ **do**
  $probability \leftarrow 1$ **for** $node \in PathToRoot(leaf)$ **do**
    $probability \leftarrow probability \cdot Visits(node)/Visits(Siblings(node))$
      `// Siblings zawiera również node`
  $movePr \leftarrow$
  $movePr \cup \{(PathToMove(PathToRoot(leaf)), probability)\}$

Wygaszanie starych wyników w drzewie **Nowe**

for $node \in Nodes$ **do**

$visits(node) \leftarrow c \cdot visits(node)$

$quality(node) \leftarrow c \cdot quality(node)$

$$c \in [0, 1]$$

Ewaluacja **Nowe**

Metoda

- Wyznacz rozkład prawdopodobieństwa ruchów obrońcy daną metodą
- Wyznacz strategię atakującego według tego rozkładu
- Oblicz wartość oczekiwaną wyniku obrońcy

!

- UCT walczy z przeciwnikiem, którego być może jeszcze nie widziało

Wnioski **Nowe**

- Model Stackelberga jest trudny dla UCT

Literatura **Nowe**

Basar, T. and G. J. Olsder (1982). *Dynamic Noncooperative Game Theory*. Academic Press Inc.

Paruchuri, Praveen et al. (2008). "Playing games for security: an efficient exact algorithm for solving Bayesian Stackelberg games". In: *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 2*. International Foundation for Autonomous Agents and Multiagent Systems, pp. 895–902.