

Dylemat więźnia

Agentowe systemy społeczne i usytuowane

Mateusz Plinta, Radosław Kazior

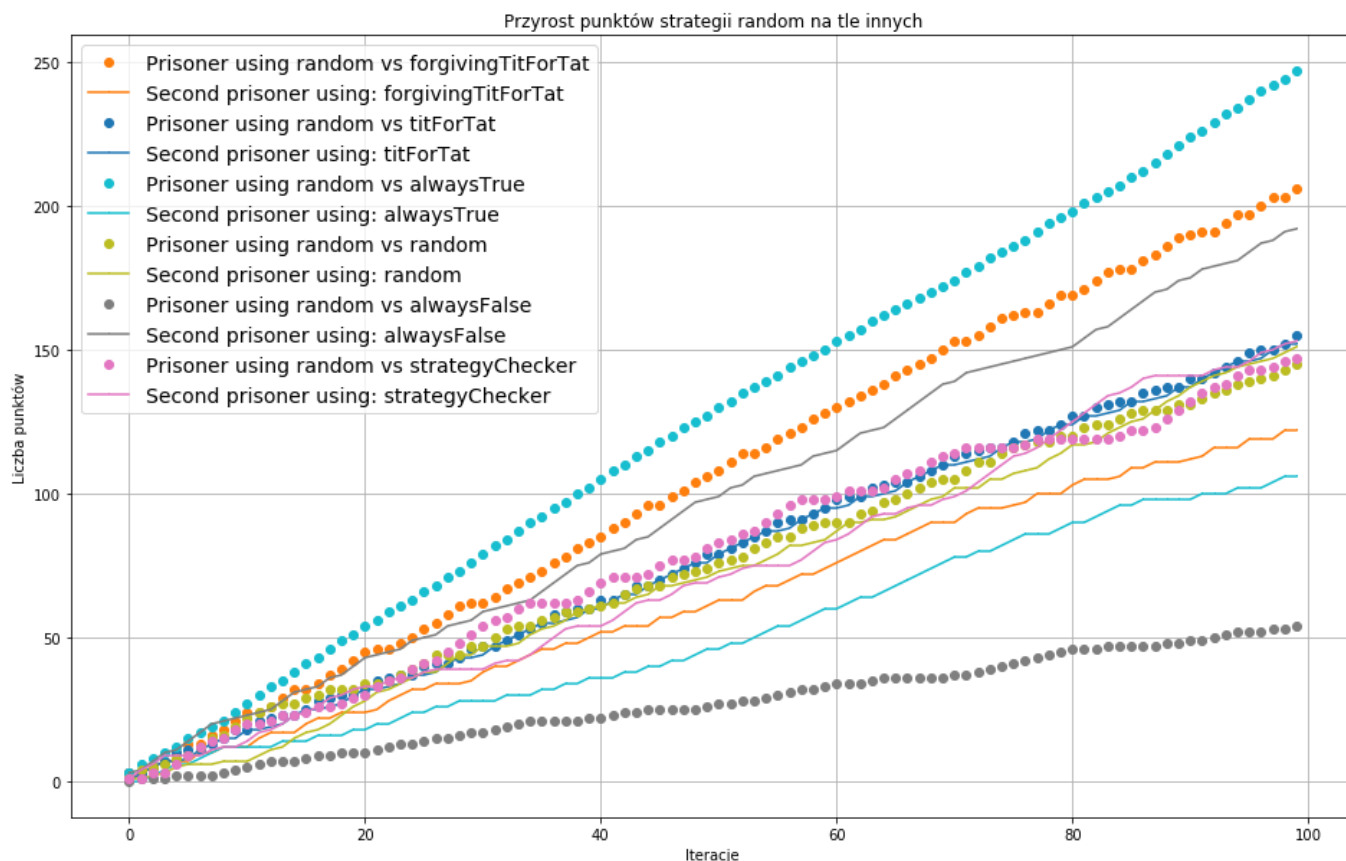
Przeprowadzono analizę sześciu strategii podejmowania decyzji w popularnej grze - dylemat więźnia. Symulację oraz strategie zaimplementowano w języku Go. Wśród analizowanych strategii można wyróżnić:

- `random` - losowanie decyzji więźnia
- `alwaysTrue` - więzień zawsze idzie na współpracę
- `alwaysFalse` - więzień zawsze zdradza
- `titForTat` - rozpocznij od współpracy, następnie kopiuj ostatni ruch drugiego więźnia
- `forgivingTitForTat` - rozpocznij od współpracy; zdradź jedynie jeśli drugi więzień zdradzi dwa razy pod rząd
- `strategyChecker` - bardziej rozbudowana strategia; rozpoczyna od sekwencji: zdradź, zdradź, współpracuj, zdradź, współpracuj. Następnie, na podstawie decyzji drugiego więźnia, podejmuje odpowiednie decyzje. Cechy strategii:
 - Próbuje nie współpracować jeśli natrafi na więźnia który nie współpracuje
 - Współpracuje gdy trafi na więźnia który chce współpracować
 - Raz współpracować i raz zdradzać gdy natrafi na więźnia który "wybacza"
 - Nie współpracować gdy trafi na więźnia który zawsze współpracuje

Czas trwania symulacji wynosił 100 iteracji. Warto zauważyć, iż więźniowie nie znają długości trwania symulacji, aby nie mogli wykorzystać tej informacji do zdobycia większej ilości punktów, np. podczas ostatniej iteracji zdradzając.

Symulację przeprowadzono 10 razy, poniżej przedstawiono 6 wykresów porównujących daną strategię ze wszystkimi innymi, na przykładzie konkretnej symulacji.

random

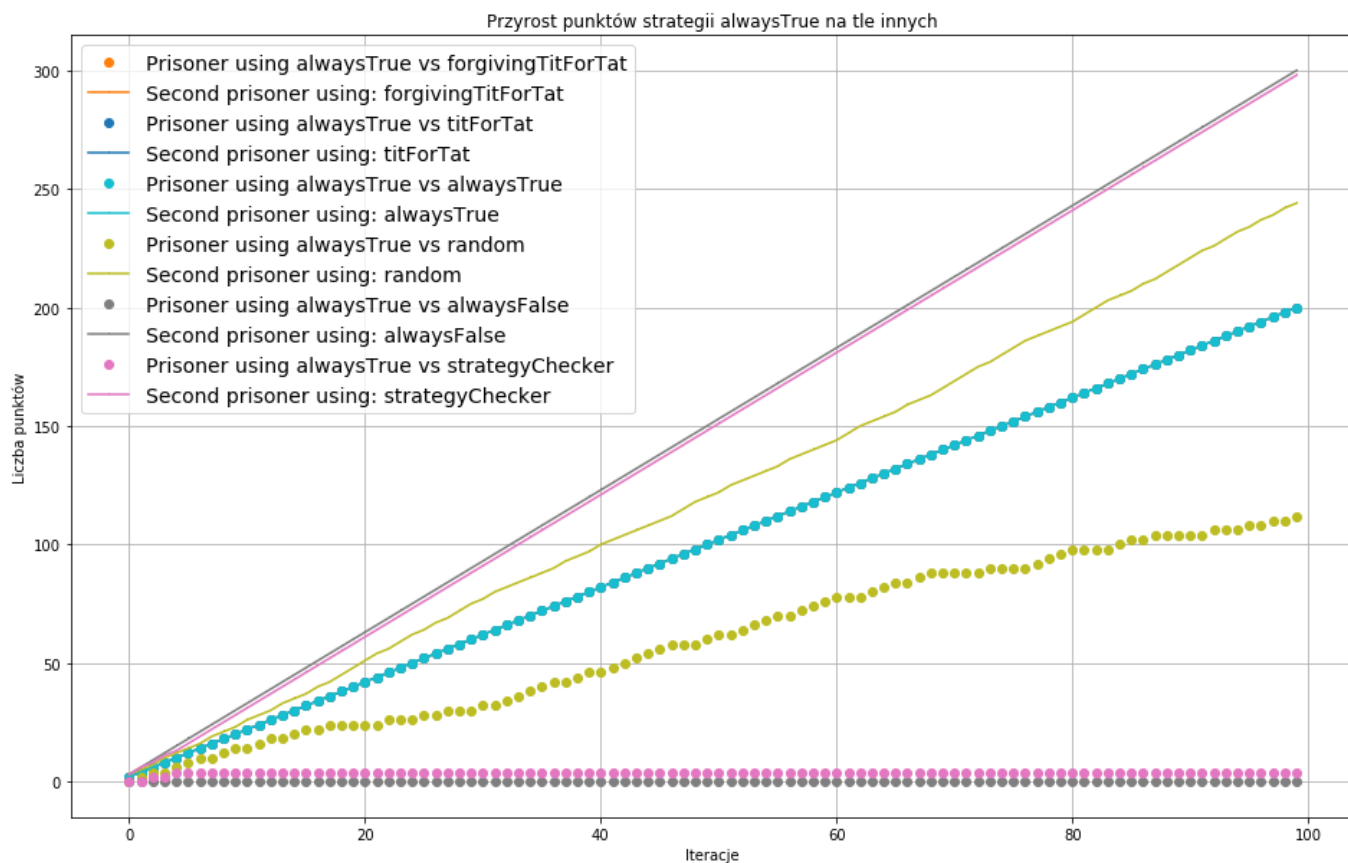


Dla porównania strategii **random** widzimy największe fluktuacje w przyroście punktów dla każdego algorytmu przeciwnika, co wynika z wbudowanej losowości tego algorytmu.

Najważniejsze wnioski z wykresu:

- używając strategii *random* największą liczbę punktów zdobędziemy grając z więźniem wykorzystującym strategię *alwaysTrue*, trochę mniej punktów zdobędziemy grając z więźniem stosującym taktykę *forgivingTitForTat*
- strategia *random* przegrywa znacząco ze *alwaysFalse*
- pozostałe warianty strategii *random* vs pozostałe mieszczą się w przedziale 100 - 150 pkt. dla końcowej iteracji

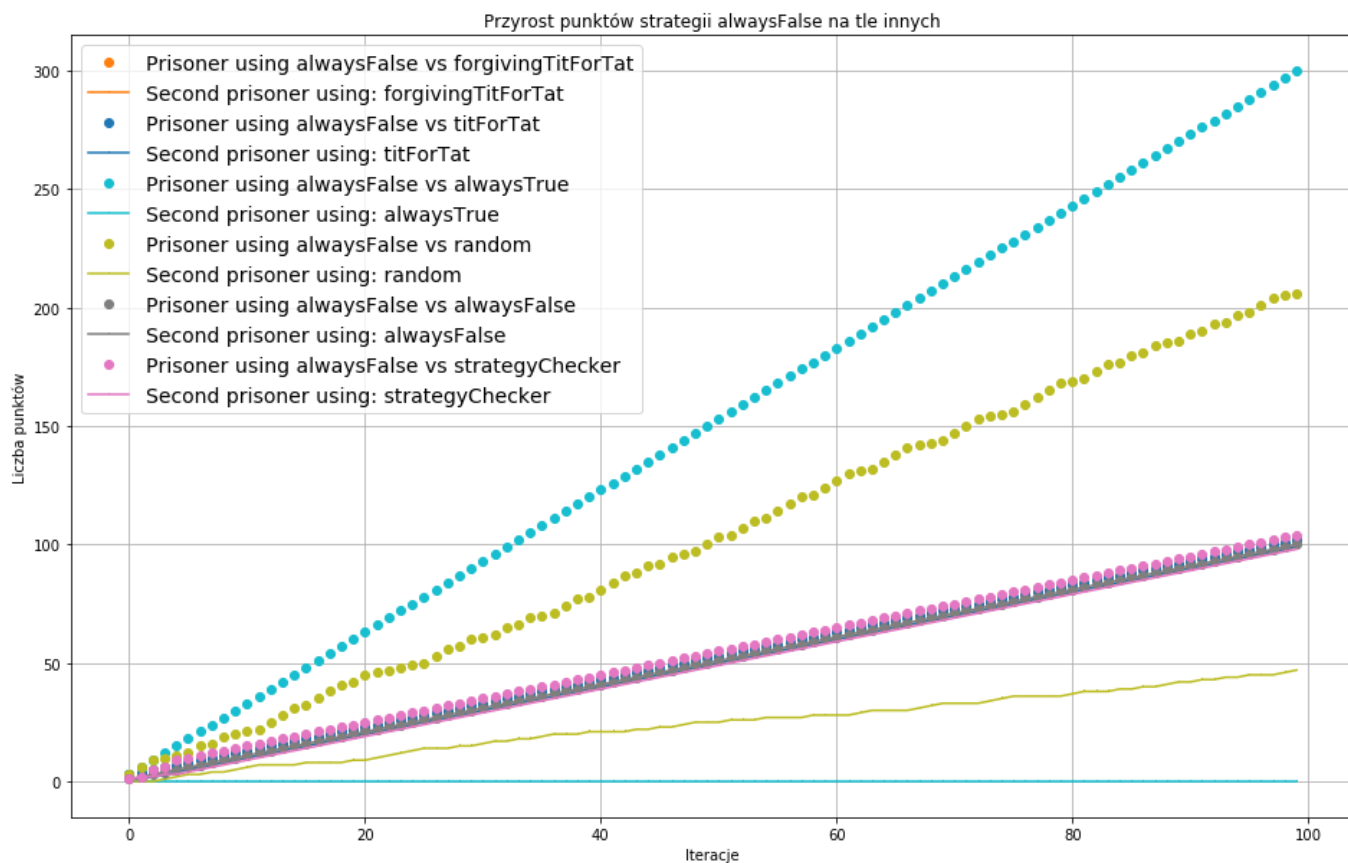
alwaysTrue



Strategia *alwaysTrue*:

- przegrywa praktycznie do zera z *alwaysFalse* oraz *strategyChecker* - przeciwnicy uzyskują nawet do 300 punktów
- *alwaysTrue* i *titForTat* zdobywają kolejno tyle samo punktów, ponieważ ciągle współpracują

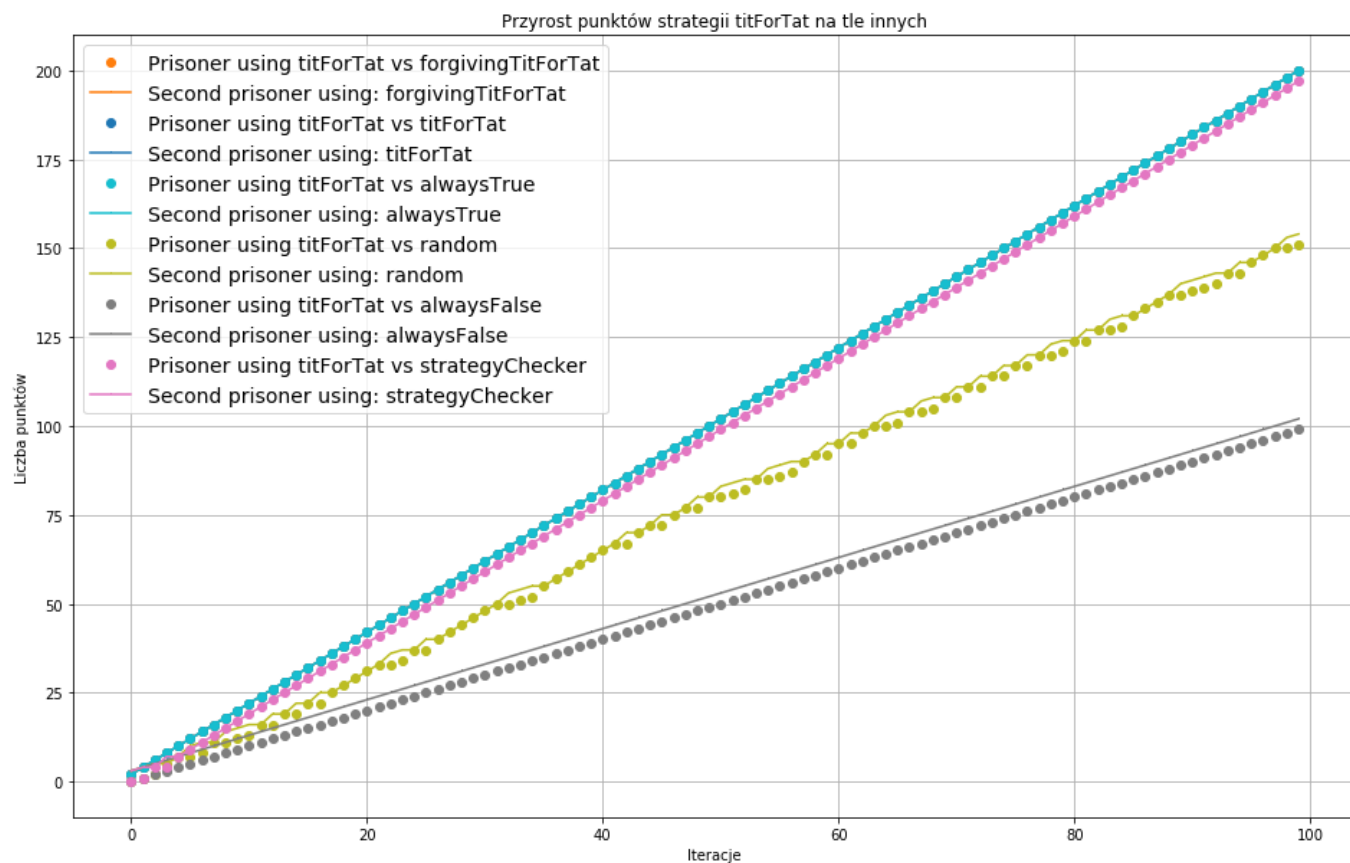
alwaysFalse



Strategia *alwaysFalse*:

- wygrywa do zera z *alwaysTrue*
- osiąga trochę gorszy wynik, ale wciąż wygrywa, ze strategią *random*
- idzie ex aequo w starciu z pozostałymi strategiami, osiągając końcowy wynik ok. 100 pkt.

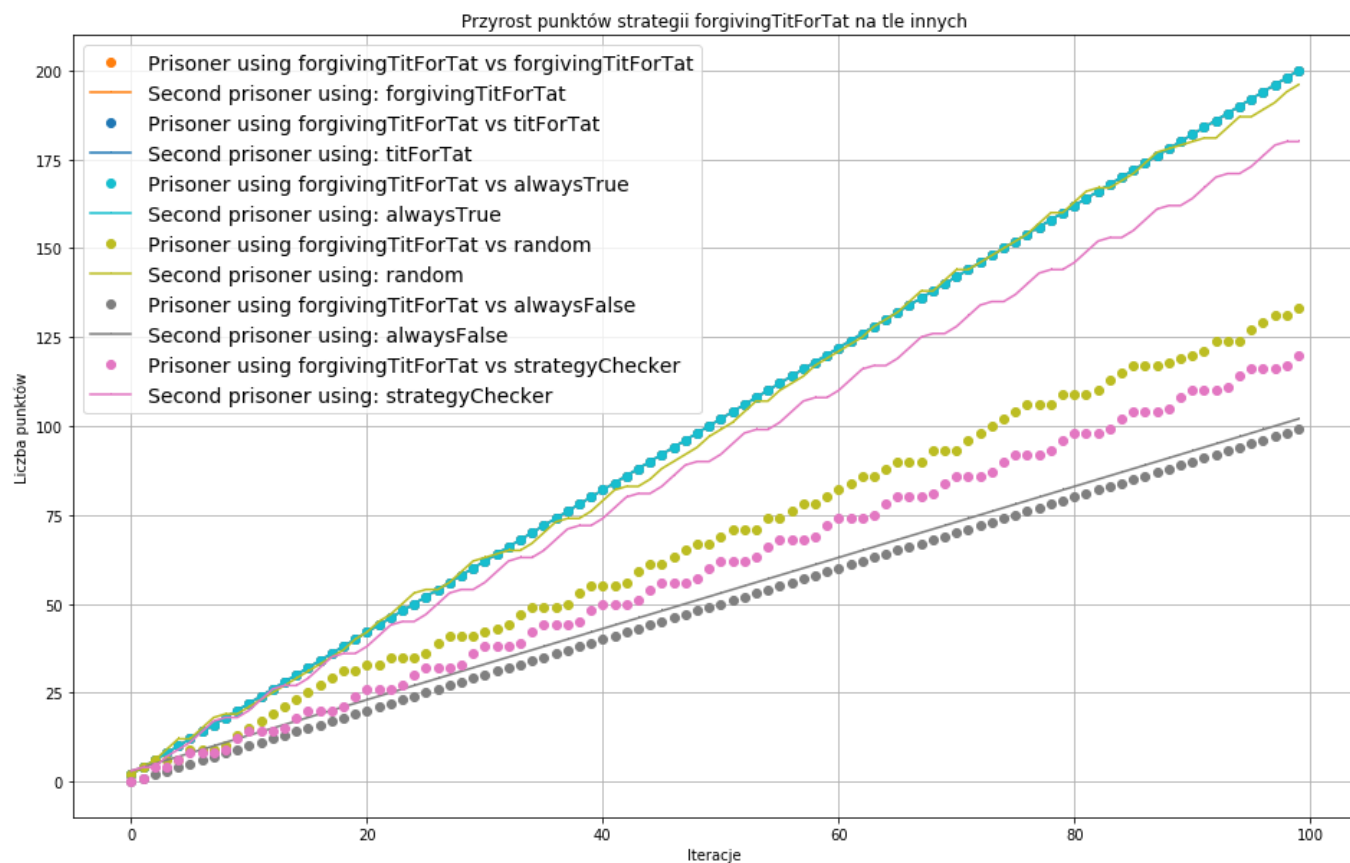
titForTat



Strategia *titForTat*:

- ex aequo w starciu z *titForTat* oraz *forgivingTitForTat*,
- vs *alwaysFalse* osiągają prawie tą samą ilość punktów, najmniejszą w porównaniu ze wszystkimi innymi strategiami, w wysokości 100 pkt. po ostatniej iteracji

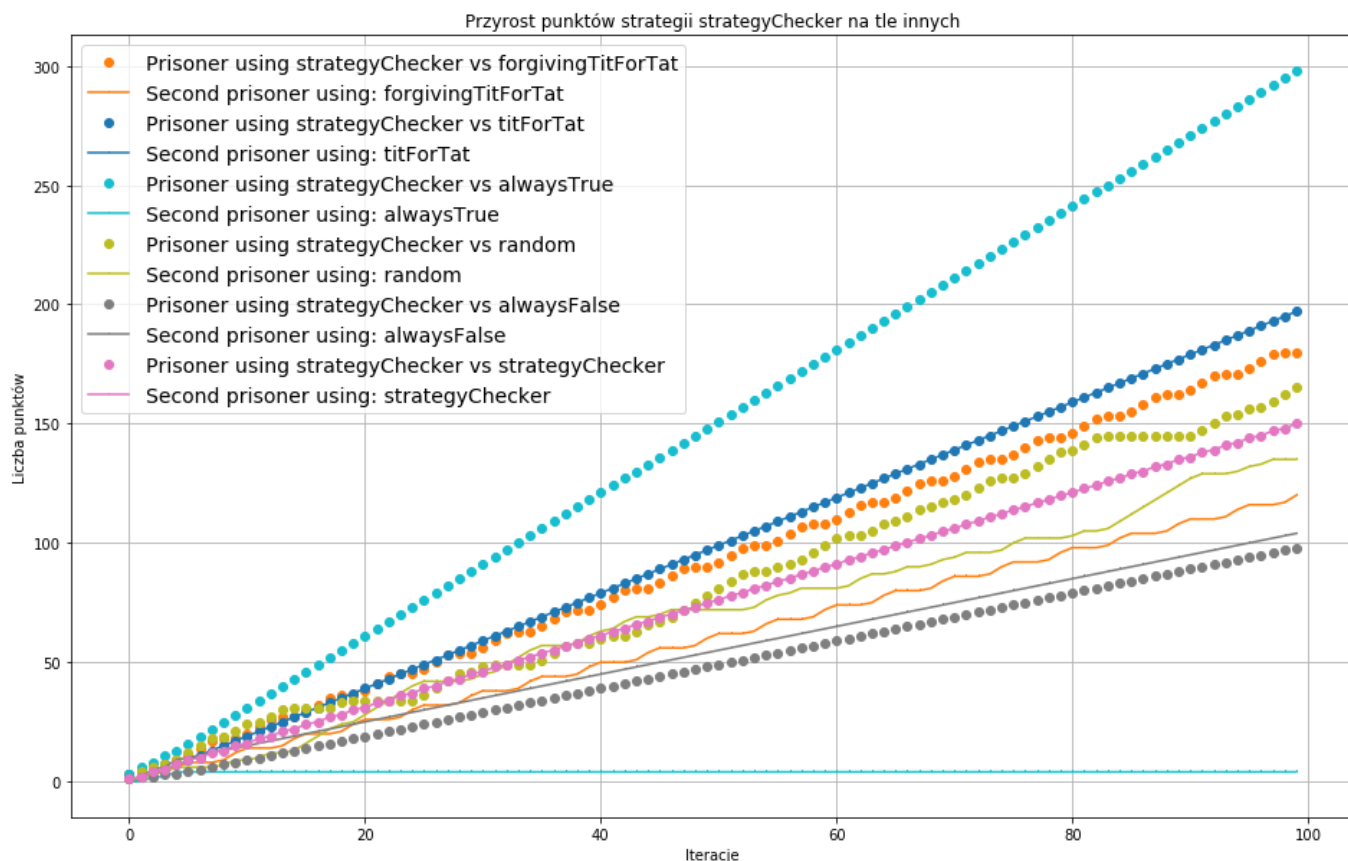
forgivingTitForTat



Strategia *forgivingTitForTat*:

- jak opisano powyżej, najwyższe wyniki uzyskuje w starciu z *titForTat*
- przegrywa tę samą ilością punktów z *random* co uzyskuje w starciu z *titForTat*, oraz trochę mniejszą ilością punktów ze *strategyChecker*

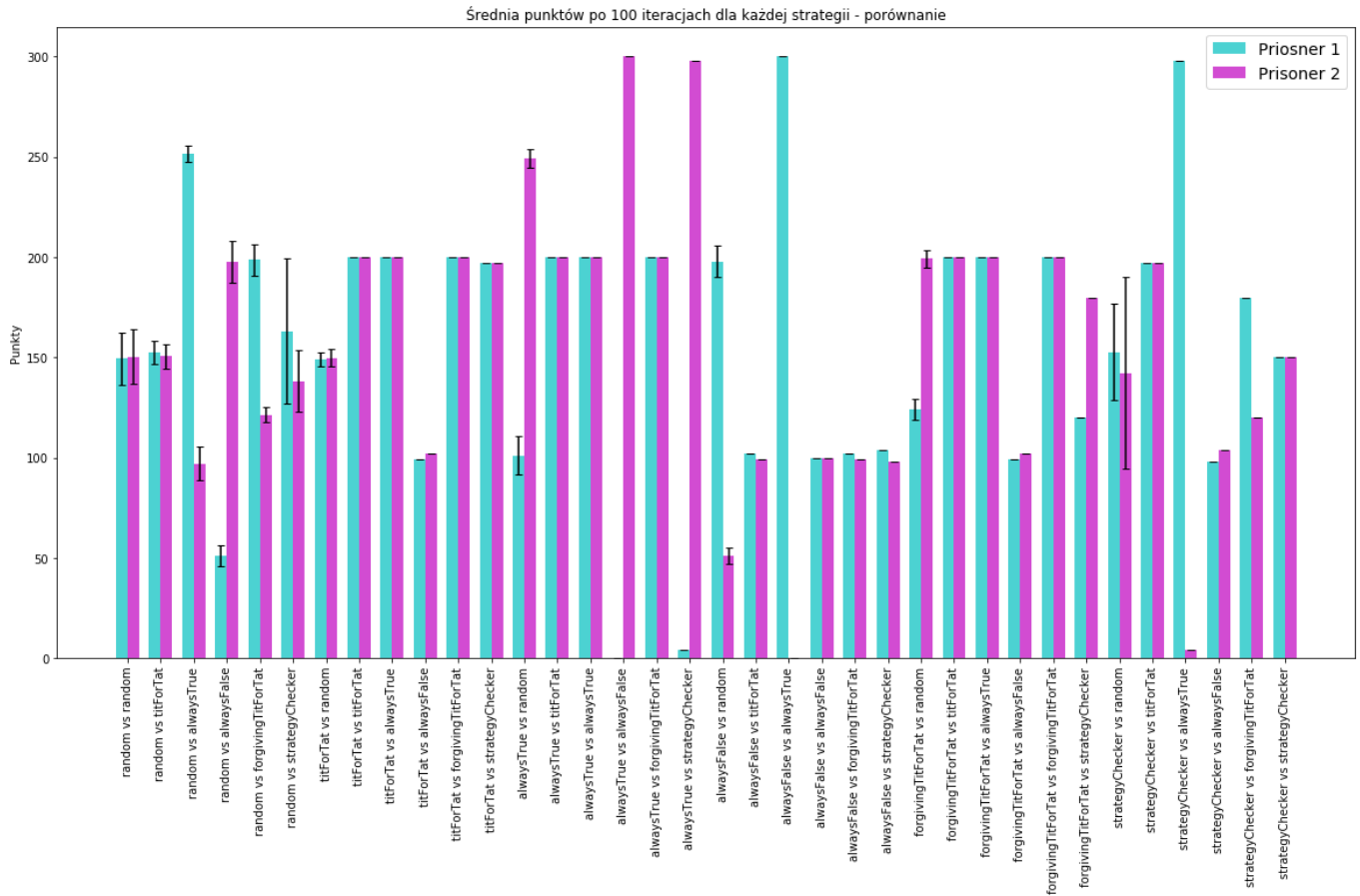
strategyChecker



Strategia *strategyChecker*:

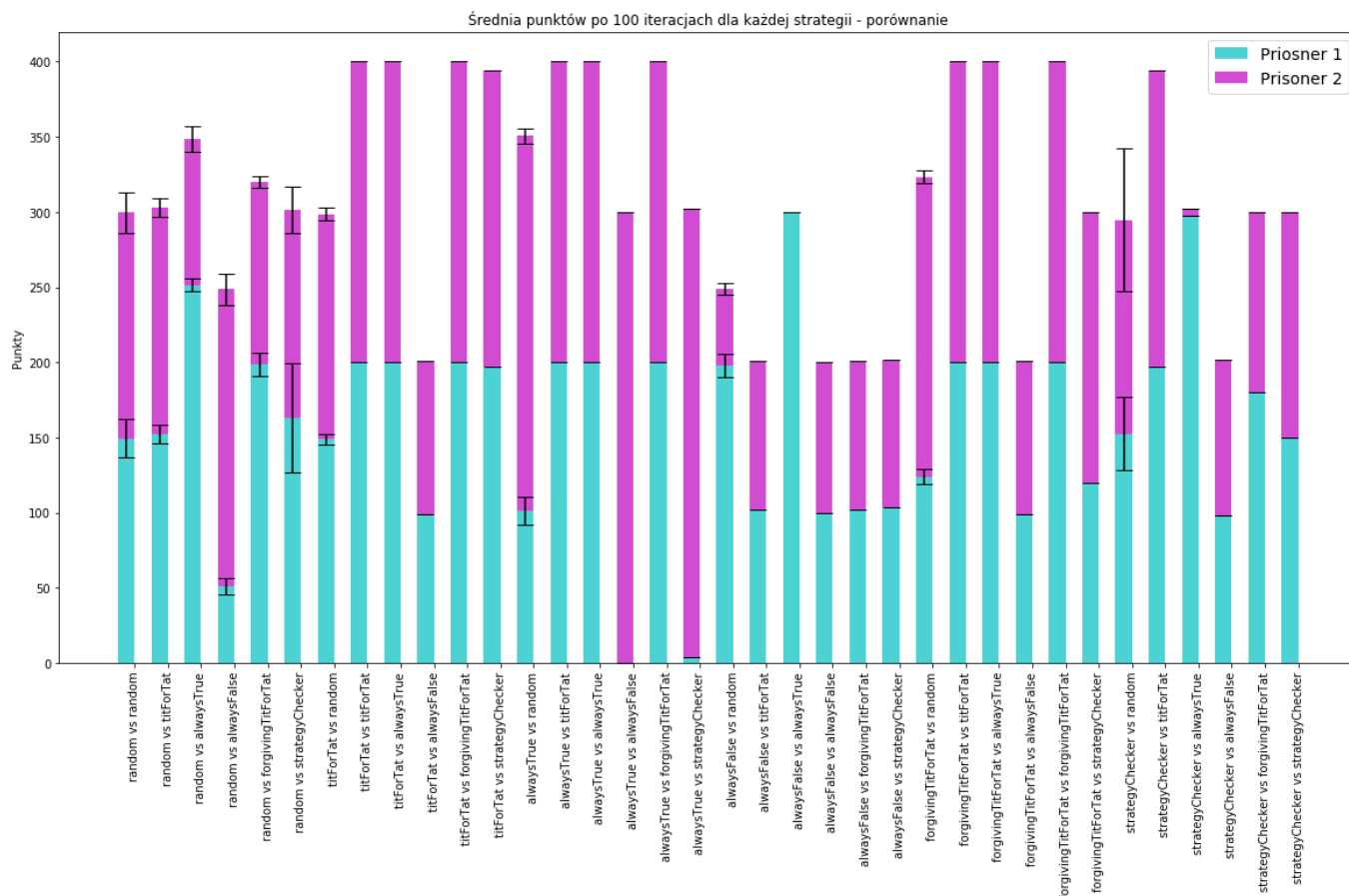
- najlepiej wypada w starciu ze *alwaysTrue*, zmiatając przeciwnika 300 pkt. do 0
- najmniej opłacalna przy walce z *alwaysFalse*, obie strony osiągną prawie 100 pkt., z niewielką przewagą przeciwnika
- **strategia ta jest najlepsza do walki praktycznie ze wszystkimi innymi zakładanymi przez nas strategiami, poza *alwaysFalse* z którym przegramy, oraz *titForTat*, z którym zostanie osiągnięty remis**

Podsumowanie - słupki



Wykresy [podsumowanie 1](#) oraz [podsumowanie 2](#) przedstawiają średnie wartości wszystkich zdobytych punktów obu więźniów po wykonaniu się ostatniej iteracji, wraz z odchyleniem standardowym tych wartości. Należy zaznaczyć, iż nie wszystkie pary strategii posiadają odchylenie standardowe, ponieważ suma zdobytych punktów jest zawsze taka sama, co wynika z braku losowości w obu algorytmach.

Podsumowanie - słupki na stosie



Wykresy różnią się jedynie położeniem słupków (w pods. 2 są na jeden nad drugim), aby zwiększyć czytelność wykresów.

Wnioski:

- najwyższe wyniki uzyskujemy wykorzystując strategię *strategyChecker* lub *alwaysFalse* w walce z *alwaysTrue* (ponieważ w przypadku gdy przeciwnik korzysta z *alwaysTrue*, *strategyChecker* właściwie staje się *alwaysFalse*)
- największe odchylenie standardowe obserwujemy w pojedynku między *strategyChecker* i *random* - obie partie osiągają porównywalne wyniki; w tym starciu nie możemy stwierdzić z jakąkolwiek pewnością o wygranej kogokolwiek, dopóki nie zakończymy symulacji
- kolejny najwyższy wynik można uzyskać stosując strategię *random* przeciwko *alwaysTrue*
- średnio podobne wyniki uzyskamy stosując poniższe kombinacje:
 - *titForTat* vs *titForTat*
 - *titForTat* vs *alwaysTrue*
 - *titForTat* vs *forgivingTitForTat*
 - *titForTat* vs *strategyChecker*
 - *alwaysTrue* vs *alwaysTrue*
 - *alwaysTrue* vs *forgivingTitForTat*
 - *forgivingTitForTat* vs *forgivingTitForTat*