

# Zagadnienie ofiary i łowcy z wykorzystaniem systemu wieloagentowego

Marek Fudaliński

Wojciech Gruszka

Wiktor Sus

8 kwietnia 2020

# 1 Wstęp

Systemy wieloagentowe to rodzaj komputerowego systemu złożonego z wielu współdziałających inteligentnych agentów. Mogą rozwiązywać problemy trudne lub niewykonalne dla indywidualnych agentów[3].

## 1.1 Opis problemu i założeń

Nasz projekt skupia się na zagadnieniu Ofiary i Łowcy, na przykład watahy wilków i stada owiec. Zadaniem łowców jest upolowanie jak największej liczby ofiar w jak najkrótszym czasie, a zadaniem ofiar jest osiągnięcie jak najdłuższego czasu życia.

Problem zamodelowaliśmy jako dwu-wymiarową siatkę, po której mogą przemieszczać się agenci (ofiary i łowcy). Każdy z agentów posiada ten sam zestaw określających go parametrów, które można jednak modyfikować osobno dla każdego z nich lub dla całej grupy. Agenci mogą przemieszczać się w czterech podstawowych kierunkach, mają ograniczone pole widzenia (1 pole przed sobą) oraz określoną prędkość, która wyraża się w ilości pól które mogą pokonać w jednym kroku symulacji.

## 2 Opis narzędzi

Przed rozpoczęciem projektu przeprowadziliśmy badania istniejących rozwiązań. Interesowały nas głównie projekty napisane w języku Python, ponieważ nasz zespół czuł się w nim najpewniej.

Ciekawym rozwiązaniem okazał się być system służący do trenowania agentów udostępniony przez grupę OpenAI o nazwie *Gym*[2]. Docelowo jednak, udostępniona biblioteka nie była projektowana z myślą o systemach wieloagentowych. Jej głównym zastosowaniem są systemy, gdzie każdy agent może pełnić dowolną funkcję, jak na przykład gry dla dwóch osób o identycznych zasadach dla obu graczy. Udało nam się jednak znaleźć wersję biblioteki OpenAi rozwiniętą o wsparcie dla wielu agentów[1]. W odróżnieniu od oryginalnej biblioteki każdy agent posiada własne parametry, co pozwala na indywidualną naukę każdego z nich. Takie rozwiązanie z kolei umożli-

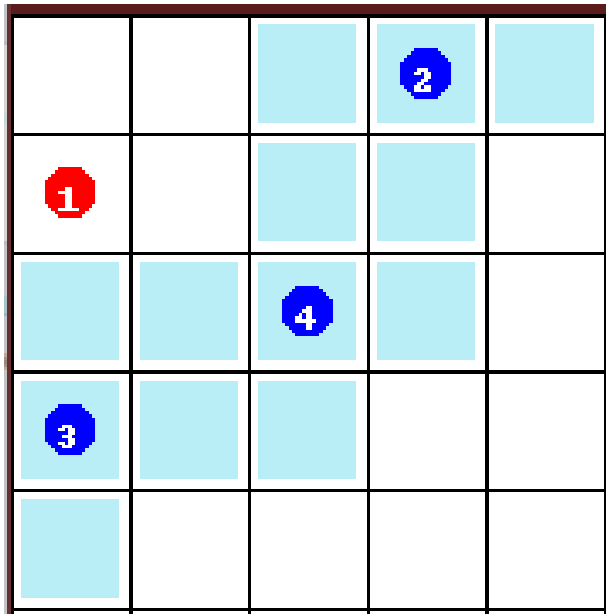
wia modelowanie bardziej zaawansowanych zachowań i systemów.

## 3 Implementacja

Używana przez nas biblioteka *MultiAgent-Gym* domyślnie dostarcza przykładowe środowiska, które reprezentują różnego rodzaju problemy (w tym przypadku są to różnego rodzaju gry, w których agenci konkurują ze sobą).

Do naszych celów konieczne było utworzenie nowego środowiska, korzystając z bazowych elementów biblioteki. Najistotniejszą zmianą jest sposób uczenia poszczególnych agentów. W przypadku domyślnego przykładu gry *Predator and Prey* agentami są jedynie łowcy, podczas gdy ofiary poruszają się losowo. Chcieliśmy jednak zamodelować sytuację dwóch konkurujących zespołów, zatem wprowadziliśmy następujące modyfikacje:

- Każdy aktor jest oddzielnym agentem, z własnym zestawem parametrów
- Każdy agent otrzymuje indywidualnie obliczaną nagrodę na podstawie określonych kryteriów
- Każdy agent komunikuje się z innymi, przekazując informacje o aktualnie obserwowanym otoczeniu, co pozwala całej grupie na podejmowanie decyzji w oparciu o wspólną wiedzę (Rys. 1).



Rysunek 1: Obszar widziany przez agenta nr 4.

Niezależnie od siebie agenci podejmują następujące decyzje:

- Wykonują optymalny ruch na podstawie obserwacji
- Uczą się na podstawie każdej podjętej akcji, obserwacji konsekwencji swoich działań przed i po wykonaniu ruchu oraz otrzymanej nagrodzie

## 4 Otrzymane wyniki

Podczas testów różnych wariantów otrzymaliśmy następujące wyniki:

- Gdy ewaluujemy wyuczonych predatrów z nieuczonymi ofiarami, wygrywają predatorzy 40% walk.
- Gdy ewaluujemy nieuczonych predatrów z wyuczonymi ofiarami, wygrywają predatorzy 13% walk.

*W trakcie*

## 5 Podsumowanie

*W trakcie*

## Bibliografia

- [1] *Projekt MultiAgent Gym*. URL: <https://github.com/koulaturag/ma-gym> (term. wiz. 2019).
- [2] *Projekt OpenAI Gym*. URL: <https://gym.openai.com/> (term. wiz. 2020).
- [3] *Systemy wieloagentowe*. URL: [https://en.wikipedia.org/wiki/Multi-agent\\_system](https://en.wikipedia.org/wiki/Multi-agent_system) (term. wiz. 2020).