

# Dokumentacja Końcowa

Uczenie maszynowe

2022Z

Jan Guziuk (310997), Jakub Romanek (311065)

29 stycznia 2023

## Spis treści

|          |  |          |
|----------|--|----------|
| <b>1</b> | <b>Krótkie streszczenie założeń z Projektu Wstępnego</b> | <b>2</b> |
| <b>2</b> | <b>Opis funkcjonalny</b>                                 | <b>3</b> |
| 2.1      | Koncepcja we wdrożeniu . . . . .                         | 3        |
| 2.2      | Działanie programu . . . . .                             | 3        |
| 2.3      | Przykładowe działanie niewytrenowanego modelu . . . . .  | 4        |
| 2.4      | Proces uczenia się agenta. . . . .                       | 5        |
| <b>3</b> | <b>Opis algorytmów oraz zbiorów danych</b>               | <b>6</b> |
| 3.1      | Algorytm . . . . .                                       | 6        |
| 3.2      | Zbiory danych . . . . .                                  | 6        |
| <b>4</b> | <b>Raport z przeprowadzonych testów oraz wnioski</b>     | <b>6</b> |
| 4.1      | Testy . . . . .  | 6        |
| 4.2      | Wnioski . . . . .  | 8        |
| <b>5</b> | <b>Opis wykorzystanych narzędzi i bibliotek</b>          | <b>8</b> |

# 1 Krótkie streszczenie założeń z Projektu Wstępnego

Tematem projektu było zaimplementowanie problemu kombinatorycznego (układanie planu lekcji) w sposób umożliwiający zastosowanie do niego metod uczenia ze wzmocnieniem. W pierwszej części postanowiliśmy rozłożyć ten skomplikowany problem na mniejsze podproblemy, w celu ułatwienia sobie implementacji wymienionych przez nas założeń.

Problem tworzenia planu lekcji można podzielić na trzy mniejsze podproblemy:

- Harmonogram dla uczniów

Podproblem harmonogramu dla uczniów dotyczy tworzenia zrównoważonego, praktycznie bezprzerwowego (czyli z jak najmniejszą ilością okienek pomiędzy zajęciami) lub kompaktowego planu opartego na programie nauczania, w którym uczeń uczestniczy. Równowaga i idealny harmonogram mówi o właściwie podzielonej liczbie godzin przedmiotów w ciągu tygodnia, podczas gdy bezprzerwow lub kompaktowy oznacza, że jak najbardziej przedmioty są planowane tak, aby minimalizować czas spędzony w szkole. Wyjątkiem są przerwy na lunch i inne krótkie przerwy pomiędzy zajęciami. Twarde ograniczenia tego podproblemu to liczba uczniów i główny harmonogram oferowanych przedmiotów z liczbą godzin w tygodniu, sal i nauczycieli.

- Przydział nauczycieli

Podproblem przydziału nauczycieli obejmuje przydzielanie nauczycieli do określonych przedmiotów biorąc pod uwagę czynniki takie jak umiejętności nauczycieli i liczba oferowanych przedmiotów. Celem jest wyrównanie obciążenia nauczycieli, gdzie obciążenie odnosi się do łącznej liczby godzin lekcyjnych przypisanych do każdego nauczyciela.

Wymagania narzucone na podproblem przydziału nauczycieli, które są wymienione poniżej:

- Nauczyciele zatrudnieni na część etatu są zobowiązani do nauczania przedmiotów, których nie mogą nauczać nauczyciele pełnoetatowi
- Jest zarówno maksymalna, jak i minimalna liczba nauczycieli, którzy mogą zostać przydzieleni do określonego przedmiotu
- Łączna liczba różnych przedmiotów nauczanych przez każdego nauczyciela nie powinna przekraczać określonej liczby. Celem tego ograniczenia jest zmniejszenie ilości czasu przygotowania dla każdego nauczyciela
- Obciążenie nauczycieli pełnoetatowych jest priorytetowe w celu wyrównania w porównaniu z nauczycielami zatrudnionymi na część etatu

Pierwsze trzy wymagania są uważane za twarde ograniczenia, które muszą zostać spełnione, podczas gdy ostatnie jest traktowane jako miękkie ograniczenie lub preferencja.

- Przypisanie sal lekcyjnych

Podproblem przydziału pomieszczeń obejmuje przydzielanie klas do sal lekcyjnych. Jedynym ograniczeniem tego problemu jest lista dostępnych sal lekcyjnych. Dla uproszczenia każda sala będzie tak samo odpowiednia do prowadzenia różnych zajęć z wyjątkiem wychowania fizycznego. W danej jednostce lekcyjnej w każdej sali lekcyjnej może przebywać jedna klasa, jeden nauczyciel oraz może być prowadzony jeden przedmiot.

- Wybór przedmiotów

Ostatecznie kolejnym "parametrem", który będzie przez nas wybrany jest przedmiot. Teoretycznie przy tworzeniu planu lekcji powinno się zwracać uwagę na to który z przedmiotów jest cięższy dla uczniów i ustawiać je bliżej środka tygodnia, a łatwiejsze przedmioty bliżej początku i końca. Oczywiście ta reguła nie jest łatwa do osiągnięcia, gdyż każdy nauczyciel ma daną dzienną ilość godzin, które spędzi w klasie. Możemy więc wywnioskować że to jak bardzo blisko będziemy śledzić tą regułę zależy od ilości nauczycieli zdolnych uczenia przedmiotów trudniejszych.

## 2 Opis funkcjonalny

### 2.1 Koncepcja we wdrożeniu

Wdrożeniu, względem koncepcji, została dla potrzeb realizacyjnych odebrana pewna złożoność problemu. Mianowicie od dłuższego czasu przy pracy nad tym projektem zastanawialiśmy się nad skalą naszego problemu. W takim problemie jak układanie planów lekcji, złożoność problemu leży w ilości możliwych kombinacji ułożenia pojedynczego planu. Mając zmienne takie jak sale, pojemność sal, możliwe terminy tygodniowo-godzinowe, przedmioty, ilość uczęszczających ludzi na dany przedmiot i nauczycieli, od razu wprowadza potencjalne konflikty w postaci:

- Czy sala może pomieścić daną ilość uczniów?
- Czy dany nauczyciel jest w stanie prowadzić przedmiot
- Czy sala nie jest zajęta przez kogoś innego?
- Czy prowadzący nie jest przypisany do dwóch różnych przedmiotów w tym samym terminie?

Biorąc te wszystkie rzeczy pod uwagę, aby stworzyć jeden plan (np. przykładowo plan w liceum dla jednej klasy) istnieje 42000 możliwych wyborów. Dlatego dla uproszczenia rozwiązania ograniczyliśmy zbiór wyborów do wyboru sali, prowadzącego oraz terminu. Oczywiście problemy, które jesteśmy w stanie przewidzieć jak najbardziej w tej skali istnieją. W razie potrzeby można łatwo zwiększyć skalę rozwiązania, kosztem czasu. W koncepcji zdecydowaliśmy się na użycie algorytmu Monte-Carlo Tree Search (MCTS), z rodziny algorytmów Model-based. Niestety ten pomysł szybko musiał być zmieniony, z samego faktu działania MCTS. W skrócie, działanie MCTS opiera się na przewidywaniu przyszłości końcowego stanu z obecnego stanu. Kiedy algorytm już wie w którą stronę iść podejmuje jeden krok w stronę najlepszego rozwiązania w danym momencie. Po zrobieniu kroku, zapomina o przewidzianej przyszłości, analizuje środowisko i przewiduje kolejne kroki. Taka forma najlepiej działa w środowiskach, gdzie środowisko jest zmieniane niezależnie od agenta po każdym jego ruchu (szachy, Go). W kontekście naszego zagadnienia, algorytm jak najbardziej dał by radę, ale przeszukiwanie wszystkich możliwych ścieżek wyborów, w środowisku, które zmienia się jedynie przez wykonywane kroki agenta, będzie bardziej przypominać algorytm, który wylicza od razu rozwiązanie, a nie sztuczną inteligencję, która poznaje środowisko i uczy się go. Dlatego więc zdecydowaliśmy się na algorytm PPO, który bardziej szczegółowo opisany został w jednej z poniższych sekcji.

### 2.2 Działanie programu

Sam program zaczyna się od definicji danych: terminów, sal, prowadzących, przedmiotów. Następnie dla potrzeb wdrożenia zostało stworzone środowisko, które będzie reprezentować tabela dwuwymiarowa, która będzie przechowywała wyżej wymienione zmienne w celu utworzenia planu. Po stworzeniu instancji środowiska, jesteśmy w stanie przypisać nasze środowisko do modelu z algorytmem PPO.

Następnie zachodzi zresetowanie środowiska, czyli ustawienie zmiennych instancji do stanów początkowych w celu przygotowania środowiska do przeprowadzenia jednej rundy uczenia. Następnie nasz program, wykonuje kroki w rundzie. Z faktu przypisywania wartości i skali rozwiązania, program wykonuje 7 kroków w rundzie. W każdym kroku rundy, program przewiduje najlepsze możliwe akcje w przestrzeni akcji, jak może podjąć analizując przestrzeń obserwacji. W naszym przypadku przestrzeń obserwacji to tablica dwuwymiarowa, która jest z każdym krokiem powoli uzupełniana. Zaś przestrzeń akcji to możliwe wybory sali, prowadzącego oraz terminu. Po wybraniu akcji i wprowadzenia tymi akcjami zmian w przestrzeni obserwacji, program jest w stanie obliczyć nagrodę jaką przydzieli na wykonany krok w rundzie.

Przypisywanie nagród jest zależne od ilości konfliktów, które powstały w danym kroku. Za każdy spowodowany konflikt w planie, agent dostaje "nagrodę" o wartości -1, jeśli w danym kroku agent nie spowoduje konfliktu jest nagradzany wartością +1. Ilość konfliktów jest obliczana poprzez funkcję **CalculateFitness**. Sprawdzane możliwe konflikty są wypisane w sekcji wyżej.

Kroki są wykonywane, dopóki tablica nie jest w pełni zapełniona. Po pełnym wypełnieniu, stan środowiska jest resetowany funkcją **ScheduleEnv.reset()**.

## 2.3 Przykładowe działanie niewytrenowanego modelu

Na obrazie poniżej, jesteśmy w stanie zobaczyć przykładowe działanie tworzenia planów z modelem, który jeszcze nie był trenowany.

```
C:\Users\jguzi\PycharmProjects\uma\venv\Scripts\python.exe C:\Users\jguzi\PycharmProjects\uma\temp.py
Using cpu device
Wrapping the env with a 'Monitor' wrapper
Wrapping the env in a DummyVecEnv.
Runda:1 Wynik kumulatywny:-9 Ilość :9
```

| Plan # | Ilość konfliktów |   |
|--------|------------------|---|
| 1      | 9                | Dz. Matematyki,Przedmiot 1,Sala 3,Prowadzący 4,Termin 1, Dz. Matematyki,Przedmiot 3,Sala 3,Prowadzący 4,Termin 2, Dz. Informatyki,Przedmiot |

| Indeks Przedmiotu | Dział           | Przedmiot (Indeks, max ilość uczniów) | Sala (Pojemność) | Prowadzący (Indeks)           | Termin (Indeks)                    |
|-------------------|-----------------|---------------------------------------|------------------|-------------------------------|------------------------------------|
| 0                 | Dz. Matematyki  | Analiza (Przedmiot 1, 25)             | Sala 3 (35)      | Mrs Jane Doe (Prowadzący 4)   | Pon,Śr,Pt 09:00 - 10:00 (Termin 1) |
| 1                 | Dz. Matematyki  | Algebra (Przedmiot 3, 25)             | Sala 3 (35)      | Mrs Jane Doe (Prowadzący 4)   | Pon,Śr,Pt 10:00 - 11:00 (Termin 2) |
| 2                 | Dz. Informatyki | Programowanie (Przedmiot 2, 35)       | Sala 4 (30)      | Dr Steve Day (Prowadzący 3)   | Pon,Śr,Pt 10:00 - 11:00 (Termin 2) |
| 3                 | Dz. Informatyki | Grafika komputerowa (Przedmiot 4, 30) | Sala 3 (35)      | Dr Steve Day (Prowadzący 3)   | Pon,Śr,Pt 10:00 - 11:00 (Termin 2) |
| 4                 | Dz. Informatyki | Bazy Danych (Przedmiot5, 35)          | Sala 1 (25)      | Dr Steve Day (Prowadzący 3)   | Pon,Śr,Pt 09:00 - 10:00 (Termin 1) |
| 5                 | Dz. Fizyki      | Dynamika (Przedmiot6, 45)             | Sala 2 (45)      | Mr. Mike Brown (Prowadzący 2) | Pon,Śr,Pt 09:00 - 10:00 (Termin 1) |
| 6                 | Dz. Fizyki      | Fizyka Kwantowa (Przedmiot7, 45)      | Sala 3 (35)      | Mrs Jane Doe (Prowadzący 4)   | Wt,Czw 10:30 - 12:00 (Termin 4)    |

```
Runda:2 Wynik kumulatywny:-10 Ilość :11
```

| Plan # | Ilość konfliktów |   |
|--------|------------------|---|
| 1      | 11               | Dz. Matematyki,Przedmiot 1,Sala 1,Prowadzący 4,Termin 1, Dz. Matematyki,Przedmiot 3,Sala 1,Prowadzący 4,Termin 2, Dz. Informatyki,Przedmiot |

| Indeks Przedmiotu | Dział           | Przedmiot (Indeks, max ilość uczniów) | Sala (Pojemność) | Prowadzący (Indeks)         | Termin (Indeks)                    |
|-------------------|-----------------|---------------------------------------|------------------|-----------------------------|------------------------------------|
| 0                 | Dz. Matematyki  | Analiza (Przedmiot 1, 25)             | Sala 1 (25)      | Mrs Jane Doe (Prowadzący 4) | Pon,Śr,Pt 09:00 - 10:00 (Termin 1) |
| 1                 | Dz. Matematyki  | Algebra (Przedmiot 3, 25)             | Sala 1 (25)      | Mrs Jane Doe (Prowadzący 4) | Pon,Śr,Pt 10:00 - 11:00 (Termin 2) |
| 2                 | Dz. Informatyki | Programowanie (Przedmiot 2, 35)       | Sala 3 (35)      | Dr Steve Day (Prowadzący 3) | Pon,Śr,Pt 10:00 - 11:00 (Termin 2) |
| 3                 | Dz. Informatyki | Grafika komputerowa (Przedmiot 4, 30) | Sala 4 (30)      | Dr James Web (Prowadzący 1) | Wt,Czw 09:00 - 10:30 (Termin 3)    |
| 4                 | Dz. Informatyki | Bazy Danych (Przedmiot5, 35)          | Sala 1 (25)      | Dr Steve Day (Prowadzący 3) | Pon,Śr,Pt 10:00 - 11:00 (Termin 2) |
| 5                 | Dz. Fizyki      | Dynamika (Przedmiot6, 45)             | Sala 1 (25)      | Dr Steve Day (Prowadzący 3) | Pon,Śr,Pt 09:00 - 10:00 (Termin 1) |
| 6                 | Dz. Fizyki      | Fizyka Kwantowa (Przedmiot7, 45)      | Sala 3 (35)      | Dr James Web (Prowadzący 1) | Wt,Czw 10:30 - 12:00 (Termin 4)    |

```
Runda:3 Wynik kumulatywny:-5 Ilość :8
```

| Plan # | Ilość konfliktów |   |
|--------|------------------|---|
| 1      | 8                | Dz. Matematyki,Przedmiot 1,Sala 4,Prowadzący 1,Termin 1, Dz. Matematyki,Przedmiot 3,Sala 3,Prowadzący 2,Termin 1, Dz. Informatyki,Przedmiot |

| Indeks Przedmiotu | Dział           | Przedmiot (Indeks, max ilość uczniów) | Sala (Pojemność) | Prowadzący (Indeks)           | Termin (Indeks)                    |
|-------------------|-----------------|---------------------------------------|------------------|-------------------------------|------------------------------------|
| 0                 | Dz. Matematyki  | Analiza (Przedmiot 1, 25)             | Sala 4 (30)      | Dr James Web (Prowadzący 1)   | Pon,Śr,Pt 09:00 - 10:00 (Termin 1) |
| 1                 | Dz. Matematyki  | Algebra (Przedmiot 3, 25)             | Sala 3 (35)      | Mr. Mike Brown (Prowadzący 2) | Pon,Śr,Pt 09:00 - 10:00 (Termin 1) |
| 2                 | Dz. Informatyki | Programowanie (Przedmiot 2, 35)       | Sala 4 (30)      | Mr. Mike Brown (Prowadzący 2) | Pon,Śr,Pt 10:00 - 11:00 (Termin 2) |
| 3                 | Dz. Informatyki | Grafika komputerowa (Przedmiot 4, 30) | Sala 4 (30)      | Mr. Mike Brown (Prowadzący 2) | Pon,Śr,Pt 10:00 - 11:00 (Termin 2) |
| 4                 | Dz. Informatyki | Bazy Danych (Przedmiot5, 35)          | Sala 4 (30)      | Dr Steve Day (Prowadzący 3)   | Pon,Śr,Pt 09:00 - 10:00 (Termin 1) |
| 5                 | Dz. Fizyki      | Dynamika (Przedmiot6, 45)             | Sala 2 (45)      | Dr James Web (Prowadzący 1)   | Wt,Czw 09:00 - 10:30 (Termin 3)    |
| 6                 | Dz. Fizyki      | Fizyka Kwantowa (Przedmiot7, 45)      | Sala 2 (45)      | Mrs Jane Doe (Prowadzący 4)   | Wt,Czw 09:00 - 10:30 (Termin 3)    |

Rysunek 1: Program losowo tworzący plan lekcji

Na zrzucie ekranu można zobaczyć, że w planach znajduje się aktualnie sporo konfliktów, natomiast przewidywaliśmy oczywiście takie działanie - naszym celem było zbudowanie podstawowego działania programu, do którego następnie "podpięliśmy" nasze AI.

## 2.4 Proces uczenia się agenta.

|                      |             |                      |             |
|----------------------|-------------|----------------------|-------------|
| rollout/             |             | rollout/             |             |
| ep_len_mean          | 7           | ep_len_mean          | 7           |
| ep_rew_mean          | -6.36       | ep_rew_mean          | 3.52        |
| time/                |             | time/                |             |
| fps                  | 1826        | fps                  | 1443        |
| iterations           | 2           | iterations           | 15          |
| time_elapsed         | 2           | time_elapsed         | 21          |
| total_timesteps      | 4096        | total_timesteps      | 30720       |
| train/               |             | train/               |             |
| approx_kl            | 0.017452441 | approx_kl            | 0.014275052 |
| clip_fraction        | 0.261       | clip_fraction        | 0.172       |
| clip_range           | 0.2         | clip_range           | 0.2         |
| entropy_loss         | -4.15       | entropy_loss         | -2.61       |
| explained_variance   | -0.0203     | explained_variance   | 0.306       |
| learning_rate        | 0.0003      | learning_rate        | 0.0003      |
| loss                 | 2.49        | loss                 | 1.48        |
| n_updates            | 10          | n_updates            | 140         |
| policy_gradient_loss | -0.0401     | policy_gradient_loss | -0.028      |
| value_loss           | 8.58        | value_loss           | 3.19        |

|                      |             |                      |             |
|----------------------|-------------|----------------------|-------------|
| rollout/             |             | rollout/             |             |
| ep_len_mean          | 7           | ep_len_mean          | 7           |
| ep_rew_mean          | -1.53       | ep_rew_mean          | 5.38        |
| time/                |             | time/                |             |
| fps                  | 1560        | fps                  | 1428        |
| iterations           | 6           | iterations           | 27          |
| time_elapsed         | 7           | time_elapsed         | 38          |
| total_timesteps      | 12288       | total_timesteps      | 55296       |
| train/               |             | train/               |             |
| approx_kl            | 0.023847908 | approx_kl            | 0.015234071 |
| clip_fraction        | 0.323       | clip_fraction        | 0.159       |
| clip_range           | 0.2         | clip_range           | 0.2         |
| entropy_loss         | -3.76       | entropy_loss         | -1.68       |
| explained_variance   | 0.101       | explained_variance   | 0.538       |
| learning_rate        | 0.0003      | learning_rate        | 0.0003      |
| loss                 | 2.01        | loss                 | 0.741       |
| n_updates            | 50          | n_updates            | 260         |
| policy_gradient_loss | -0.0499     | policy_gradient_loss | -0.023      |
| value_loss           | 5.35        | value_loss           | 2.07        |

|                      |             |                      |             |
|----------------------|-------------|----------------------|-------------|
| rollout/             |             | rollout/             |             |
| ep_len_mean          | 7           | ep_len_mean          | 7           |
| ep_rew_mean          | 2.07        | ep_rew_mean          | 7           |
| time/                |             | time/                |             |
| fps                  | 1486        | fps                  | 1327        |
| iterations           | 10          | iterations           | 151         |
| time_elapsed         | 13          | time_elapsed         | 232         |
| total_timesteps      | 20480       | total_timesteps      | 309248      |
| train/               |             | train/               |             |
| approx_kl            | 0.016629774 | approx_kl            | 0.008132539 |
| clip_fraction        | 0.197       | clip_fraction        | 0.0948      |
| clip_range           | 0.2         | clip_range           | 0.2         |
| entropy_loss         | -3.15       | entropy_loss         | -0.436      |
| explained_variance   | 0.115       | explained_variance   | 0.995       |
| learning_rate        | 0.0003      | learning_rate        | 0.0003      |
| loss                 | 2.08        | loss                 | -0.00119    |
| n_updates            | 90          | n_updates            | 1500        |
| policy_gradient_loss | -0.0313     | policy_gradient_loss | -0.00855    |
| value_loss           | 3.8         | value_loss           | 0.0187      |

Rysunek 2: Proces uczenia się agenta

Powyżej widać poszczególne wycinki z logów uczenia się agenta. Jak jesteśmy w stanie zobaczyć na początku agent miał medianę wyniku równą **-6.36** (ta wartość jest średnią nagród jakie dostawał agent podczas podejmowania się akcji na środowisku). Dostajemy dodatkowe informacje na temat ilości wykonanych rund (total.timesteps 4096) w danym momencie oraz szybkości ich wykonania (fps (frames per second) 1826).

Dodatkowo są zdefiniowane:

- `approx_kl` - wartość różnicy pomiędzy poprzednią a nową powstałą polityką
- `clip_range` - hiperparametr dla przechodzenia do docelowej polityki. Czyli jak daleko nowa polityka może się oddalić od starej polityki, a jednocześnie nadal przynieść zyski
- `explained_variance` - wariancja, którą możemy interpretować jako stabilność modelu tzn. jak bardzo przewidywalne są wyniki za możliwe akcje agenta.
- `n_updates` - ilość aktualizacji gradientów

Jak jesteśmy w stanie zaobserwować, wraz z rozwojem mediana otrzymywanej nagrody zbliża się do granicy 7. Jest tak gdyż, maksymalną wartość nagrody jaką może uzyskać agent w danej rundzie jest równa ilości kroków, pomnożona razy maksymalną nagrodę, czyli w skrócie  $7 * 1 = 7$ .

## 3 Opis algorytmów oraz zbiorów danych

### 3.1 Algorytm

Algorytmem, z którego ostatecznie skorzystaliśmy, jest algorytm **Proximal Policy Optimization**. PPO to rodzina bezmodelowych algorytmów uczenia ze wzmocnieniem opracowanych przez OpenAI. Algorytmy PPO są metodami gradientowej polityki, co oznacza, że przeszukują przestrzeń polityk, zamiast przypisywać wartości parom stan-akcja. Algorytmy PPO mają niektóre z zalet algorytmów **trust region policy optimization (TRPO)**, ale są prostsze w implementacji, bardziej ogólne i mają lepszą złożoność próbkową. Jednak uzyskanie dobrych wyników przy użyciu metod gradientowej polityki jest trudne, ponieważ są one wrażliwe na wybór kroku - zbyt mały, a postęp jest beznadziejnie powolny; zbyt duży i sygnał jest zdominowany przez szum lub można zobaczyć katastrofalne spadki wydajności. Często również mają bardzo słabą wydajność wzorców, wymagając milionów (lub miliardów) kroków czasowych, aby nauczyć się prostych zadań. Droga do sukcesu w uczeniu ze wzmocnieniem nie jest tak oczywista - algorytmy mają wiele ruchomych części, które trudno jest debugować i wymagają znacznego wysiłku w dostrajaniu, aby uzyskać dobre wyniki. PPO łączy łatwość implementacji, złożoność wzorca i łatwość dostrajania, próbując obliczyć aktualizację w każdym kroku, która minimalizuje funkcję kosztu, jednocześnie zapewniając, że odchylenie od poprzedniej polityki jest stosunkowo małe.

### 3.2 Zbiory danych

W przypadku Reinforcement Learning trudno jest zdefiniować zbiór trenujący, gdyż jego celem nie jest aproksymowanie pewnego nieznanego odwzorowania przez generalizację na podstawie zbioru przykładów trenujących. Systemowi uczącemu się ze wzmocnieniem nie są dostarczane żadne przykłady trenujące, a jedynie wartościująca informacja trenująca, oceniająca jego dotychczasową skuteczność. Tą wartościującą informacją trenującą są nasze twarde i miękkie ograniczenia, które jeśli nasz algorytm im bliżej będzie się trzymał tym większą nagrodę dostanie, a im bardziej będzie odchodził od reguł tym mniejszą nagrodę dostanie. Oczywiście zbiorem danych w naszym przypadku możemy też nazwać zbiór wartości akcji, które są podejmowane przez algorytm. Ten zbiór danych zawiera wartości każdej akcji, którą może wykonać algorytm w danym momencie. Wartości tych akcji determinują, jakie akcje podejmuje algorytm.

## 4 Raport z przeprowadzonych testów oraz wnioski

### 4.1 Testy

Po podpięciu sztucznej inteligencji do naszego programu, wykonaliśmy szereg testów, w celu sprawdzenia czy wszystko przebiega tak, jak byśmy tego oczekiwali. Mianowicie, w kolejnych iteracjach algorytmu, powinniśmy zobaczyć tendencję wzrostową nagrody względem tworzonego planu. Ku naszemu zaskoczeniu, algorytm zaczął działać za pierwszym razem i przynosić ciekawe rezultaty. Jak widać poniżej, po wykonaniu 350000 kroków, nasz program jest w stanie wyprodukować plany, które nie posiadają w sobie konfliktów.

| Runda:1 Wynik kumulatywny:7 Ilość :0 |                  |   |                  |                             |                                    |
|--------------------------------------|------------------|---|------------------|-----------------------------|------------------------------------|
| Plan #                               | Ilość konfliktów |   |                  |                             |                                    |
| 1                                    | 0                | Dz. Matematyki,Przedmiot 1,Sala 1,Prowadzący 1,Termin 4, Dz. Matematyki,Przedmiot 3,Sala 4,Prowadzący 1,Termin 1, Dz. Informatyki,Przedmiot |                  |                             |                                    |
| Indeks Przedmiotu                    | Dział            | Przedmiot (Indeks, max ilość uczniów)   | Sala (Pojemność) | Prowadzący (Indeks)         | Termin (Indeks)                    |
| 0                                    | Dz. Matematyki   | Analiza (Przedmiot 1, 25)   | Sala 1 (25)      | Dr James Web (Prowadzący 1) | Wt,Czw 10:30 - 12:00 (Termin 4)    |
| 1                                    | Dz. Matematyki   | Algebra (Przedmiot 3, 25)   | Sala 4 (30)      | Dr James Web (Prowadzący 1) | Pon,Śr,Pt 09:00 - 10:00 (Termin 1) |
| 2                                    | Dz. Informatyki  | Programowanie (Przedmiot 2, 35)   | Sala 3 (35)      | Dr James Web (Prowadzący 1) | Wt,Czw 09:00 - 10:30 (Termin 3)    |
| 3                                    | Dz. Informatyki  | Grafika komputerowa (Przedmiot 4, 30)   | Sala 4 (30)      | Dr Steve Day (Prowadzący 3) | Wt,Czw 10:30 - 12:00 (Termin 4)    |
| 4                                    | Dz. Informatyki  | Bazy Danych (Przedmiot5, 35)  | Sala 3 (35)      | Mrs Jane Doe (Prowadzący 4) | Wt,Czw 10:30 - 12:00 (Termin 4)    |
| 5                                    | Dz. Fizyki       | Dynamika (Przedmiot6, 45)   | Sala 2 (45)      | Dr Steve Day (Prowadzący 3) | Pon,Śr,Pt 10:00 - 11:00 (Termin 2) |
| 6                                    | Dz. Fizyki       | Fizyka Kwantowa (Przedmiot7, 45)  | Sala 2 (45)      | Mrs Jane Doe (Prowadzący 4) | Pon,Śr,Pt 09:00 - 10:00 (Termin 1) |
| Runda:2 Wynik kumulatywny:7 Ilość :0 |                  |   |                  |                             |                                    |
| Plan #                               | Ilość konfliktów |   |                  |                             |                                    |
| 1                                    | 0                | Dz. Matematyki,Przedmiot 1,Sala 1,Prowadzący 1,Termin 4, Dz. Matematyki,Przedmiot 3,Sala 4,Prowadzący 1,Termin 1, Dz. Informatyki,Przedmiot |                  |                             |                                    |
| Indeks Przedmiotu                    | Dział            | Przedmiot (Indeks, max ilość uczniów)   | Sala (Pojemność) | Prowadzący (Indeks)         | Termin (Indeks)                    |
| 0                                    | Dz. Matematyki   | Analiza (Przedmiot 1, 25)   | Sala 1 (25)      | Dr James Web (Prowadzący 1) | Wt,Czw 10:30 - 12:00 (Termin 4)    |
| 1                                    | Dz. Matematyki   | Algebra (Przedmiot 3, 25)   | Sala 4 (30)      | Dr James Web (Prowadzący 1) | Pon,Śr,Pt 09:00 - 10:00 (Termin 1) |
| 2                                    | Dz. Informatyki  | Programowanie (Przedmiot 2, 35)   | Sala 3 (35)      | Dr James Web (Prowadzący 1) | Wt,Czw 09:00 - 10:30 (Termin 3)    |
| 3                                    | Dz. Informatyki  | Grafika komputerowa (Przedmiot 4, 30)   | Sala 4 (30)      | Dr Steve Day (Prowadzący 3) | Wt,Czw 10:30 - 12:00 (Termin 4)    |
| 4                                    | Dz. Informatyki  | Bazy Danych (Przedmiot5, 35)  | Sala 3 (35)      | Mrs Jane Doe (Prowadzący 4) | Wt,Czw 10:30 - 12:00 (Termin 4)    |
| 5                                    | Dz. Fizyki       | Dynamika (Przedmiot6, 45)   | Sala 2 (45)      | Dr Steve Day (Prowadzący 3) | Pon,Śr,Pt 10:00 - 11:00 (Termin 2) |
| 6                                    | Dz. Fizyki       | Fizyka Kwantowa (Przedmiot7, 45)  | Sala 2 (45)      | Mrs Jane Doe (Prowadzący 4) | Pon,Śr,Pt 09:00 - 10:00 (Termin 1) |
| Runda:3 Wynik kumulatywny:7 Ilość :0 |                  |   |                  |                             |                                    |
| Plan #                               | Ilość konfliktów |   |                  |                             |                                    |
| 1                                    | 0                | Dz. Matematyki,Przedmiot 1,Sala 1,Prowadzący 1,Termin 4, Dz. Matematyki,Przedmiot 3,Sala 4,Prowadzący 1,Termin 1, Dz. Informatyki,Przedmiot |                  |                             |                                    |
| Indeks Przedmiotu                    | Dział            | Przedmiot (Indeks, max ilość uczniów)   | Sala (Pojemność) | Prowadzący (Indeks)         | Termin (Indeks)                    |
| 0                                    | Dz. Matematyki   | Analiza (Przedmiot 1, 25)   | Sala 1 (25)      | Dr James Web (Prowadzący 1) | Wt,Czw 10:30 - 12:00 (Termin 4)    |
| 1                                    | Dz. Matematyki   | Algebra (Przedmiot 3, 25)   | Sala 4 (30)      | Dr James Web (Prowadzący 1) | Pon,Śr,Pt 09:00 - 10:00 (Termin 1) |
| 2                                    | Dz. Informatyki  | Programowanie (Przedmiot 2, 35)   | Sala 3 (35)      | Dr James Web (Prowadzący 1) | Wt,Czw 09:00 - 10:30 (Termin 3)    |
| 3                                    | Dz. Informatyki  | Grafika komputerowa (Przedmiot 4, 30)   | Sala 4 (30)      | Dr Steve Day (Prowadzący 3) | Wt,Czw 10:30 - 12:00 (Termin 4)    |
| 4                                    | Dz. Informatyki  | Bazy Danych (Przedmiot5, 35)  | Sala 3 (35)      | Mrs Jane Doe (Prowadzący 4) | Pon,Śr,Pt 10:00 - 11:00 (Termin 2) |
| 5                                    | Dz. Fizyki       | Dynamika (Przedmiot6, 45)   | Sala 2 (45)      | Dr Steve Day (Prowadzący 3) | Pon,Śr,Pt 10:00 - 11:00 (Termin 2) |
| 6                                    | Dz. Fizyki       | Fizyka Kwantowa (Przedmiot7, 45)  | Sala 2 (45)      | Mrs Jane Doe (Prowadzący 4) | Pon,Śr,Pt 09:00 - 10:00 (Termin 1) |
| Runda:4 Wynik kumulatywny:7 Ilość :0 |                  |   |                  |                             |                                    |
| Plan #                               | Ilość konfliktów |   |                  |                             |                                    |
| 1                                    | 0                | Dz. Matematyki,Przedmiot 1,Sala 1,Prowadzący 1,Termin 4, Dz. Matematyki,Przedmiot 3,Sala 4,Prowadzący 1,Termin 1, Dz. Informatyki,Przedmiot |                  |                             |                                    |

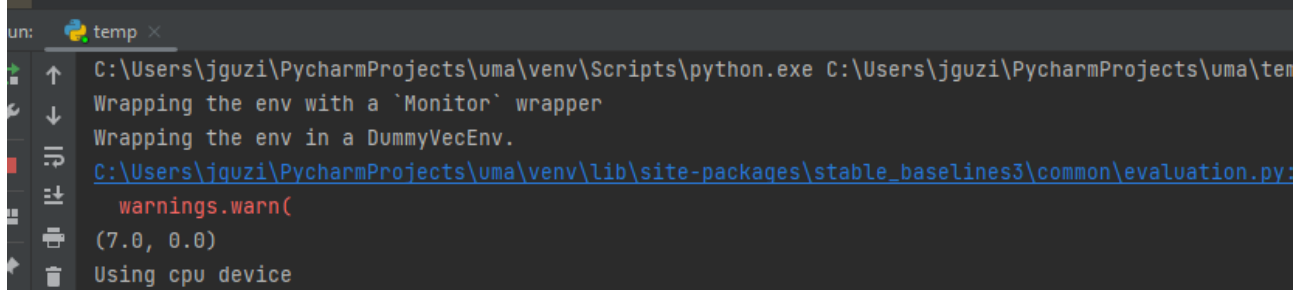
Rysunek 3: Wytrenowany program tworzący plany

Dodatkowo, biblioteka **Stablebaselines** posiada funkcję **evaluate\_policy()**, którą jesteśmy w stanie prze-testować nasz model po trenowaniu.

```

388
389     PP0_path = os.path.join('Training', 'Saved Models', 'PP0_model')
390
391     model = PP0.load(PP0_path, env=env)
392     print(evaluate_policy(model, env, n_eval_episodes=8000, render=False))

```



```

C:\Users\jguzi\PycharmProjects\uma\venv\Scripts\python.exe C:\Users\jguzi\PycharmProjects\uma\tem
Wrapping the env with a 'Monitor' wrapper
Wrapping the env in a DummyVecEnv.
C:\Users\jguzi\PycharmProjects\uma\venv\lib\site-packages\stable_baselines3\common\evaluation.py:
warnings.warn(
(7.0, 0.0)
Using cpu device

```

Rysunek 4: Wytrenowany program tworzący plany

Funkcja **evaluate\_policy()**, zwraca dwie wartości na podstawie próbki 8000 wykonanych rund. Pierwsza wartość reprezentuje medianę nagrody rundy, a druga standardowe odchylenie nagrody, w naszym przypadku jest równa zero.

## 4.2 Wnioski

Ostatecznie, mimo okrojenia funkcjonalności naszego programu względem początkowego założenia uważamy ten projekt za sukces, ponieważ główny cel został spełniony, to znaczy przez algorytm sztucznej inteligencji tworzony jest plan, który nie posiada żadnych konfliktów. Agent nauczył się tego od zera, robiąc na początku dużo pomyłek, jednak po kilkudziesięciu iteracjach tworzył w pełni poprawne plany.

W trakcie wykonywania tego projektu, nauczyliśmy się mnóstwo o uczeniu ze wzmocnieniem, obejrzelśmy bardzo dużo materiałów na ten temat, zanim w ogóle mogliśmy podjąć się wykonania praktycznej części zadania. Oczywiście, projekt ma solidne podstawy do dalszego rozwoju, na przykład dodanie większej liczby klas, co drastycznie zwiększyło by złożoność obliczeniową problemu, ale dzięki temu rozwiązanie stałoby się bardziej praktyczne.

## 5 Opis wykorzystanych narzędzi i bibliotek

Wykorzystane narzędzia i biblioteki:

- prettytable

Biblioteka wyświetlająca przekazane dane w formie tabeli, czytelnej i jasnej dla użytkownika. Wykorzystanie jej było oczywiste biorąc pod uwagę temat naszego projektu.

- numpy

Biblioteka używana do pracy z listami. NumPy używa obiektów, które są 50 razy szybsze od zwykłych list. Biblioteka używana jest w dziedzinie data science do procesowania dużych ilości danych w krótkim czasie. Jej użycie uzasadniamy potrzebą stosowania szybkich obliczeń przeprowadzanych przez nasz algorytm w trakcie kilkudziesięciu bądź nawet kilkuset generacji.

- gym

Gym jest open source'ową biblioteką Pythona służącą do tworzenia i porównywania algorytmów uczenia ze wzmocnieniem poprzez zapewnienie standardowego API do komunikacji pomiędzy algorytmami uczenia się i środowiskami, jak również standardowy zestaw środowisk zgodnych z tym API. Od momentu wydania, API gym stało się standardem w tej dziedzinie.

- stable.baselines3

Stable-Baselines3 to biblioteka umożliwiająca łatwe i szybkie tworzenie i uczenie modeli uczenia maszynowego dla szerokiego zakresu problemów związanych z sztuczną inteligencją. Jest oparta na popularnej bibliotece OpenAI Gym i TensorFlow i zawiera wiele gotowych algorytmów uczenia, takich jak A2C, PPO, DQN itp. Biblioteka jest łatwa w użyciu i zaprojektowana tak, aby zminimalizować ilość kodu wymaganego do uruchomienia. Do naszego projektu skorzystaliśmy konkretnie z rodziny algorytmów PPO, która została opisana w sekcji powyżej.