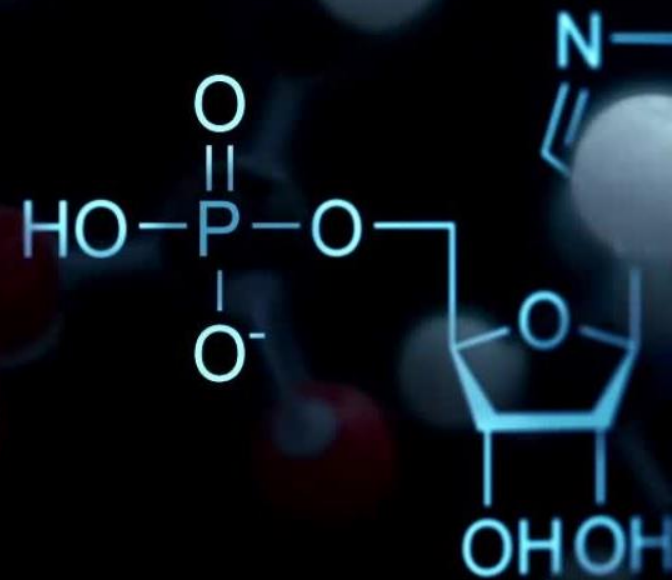


Algorytmy genetyczne, a "rozwiązanie" autorskiej gry "Bitwa"

Marcin Wojciechowski

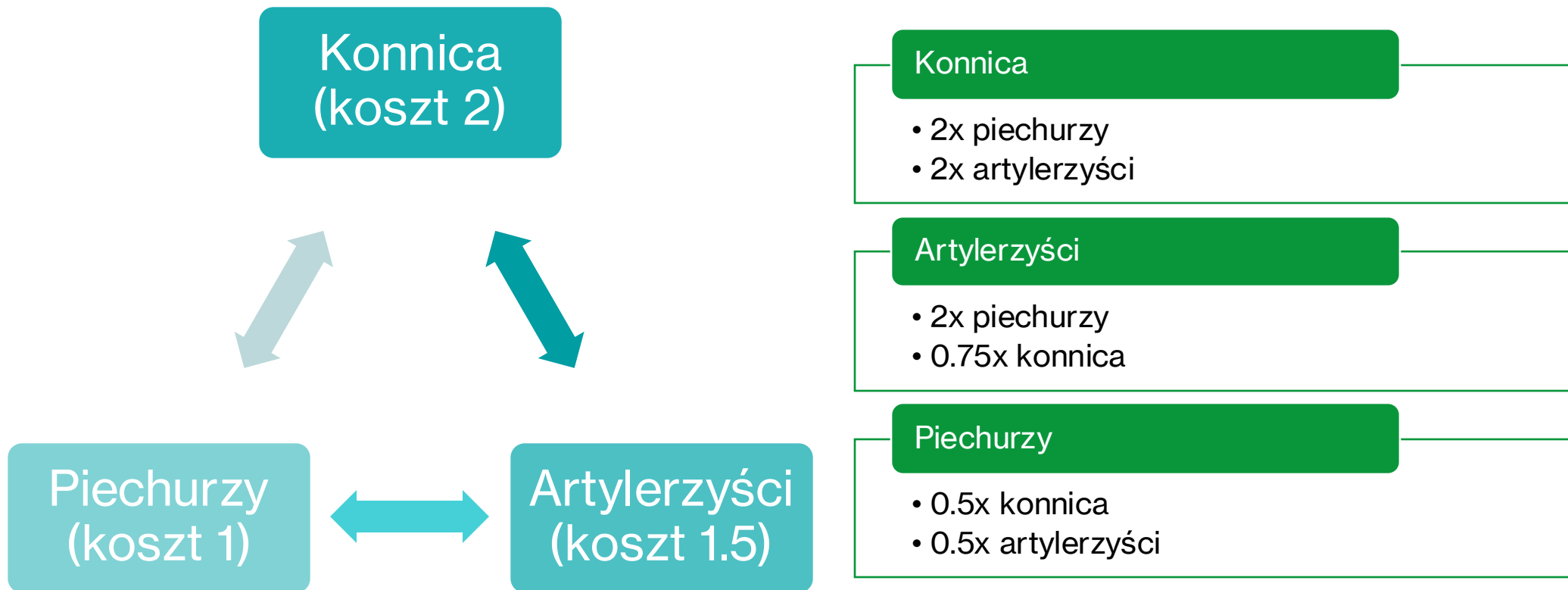


Repo + bibliografia

- Repozytorium: <https://gitlab.com/mwojciechowski653/io-projekt-2>
- Dokumentacja PyGAD(11-13.06): <https://pygad.readthedocs.io/en/latest/>
- Niebinarność genów(12.06): <https://blog.paperspace.com/working-with-different-genetic-algorithm-representations-python/>
- Różne sposoby krzyżowania genów(12-13.06): <https://blog.derlin.ch/genetic-algorithms-with-pygad>



Założenia gry



Baza danych

Autorska, stworzyłem własny generator

6 kolumn, 3x według schematu: typ jednostki, jej liczebność

5000 rekordów, funkcjonalność w *main.py* ustawiona na dodanie dodatkowego 1000 za każdym razem

Budżet zarówno armii wroga, jak i naszej to tym razem 1000

Geny

6 genów niebinarnych z przedziału $<0, 1>$ każdy. Dla genów 1., 3. i 5. interpretowane są one odpowiednio jako typ jednostki:

- 1 – piechurzy, dla $0 \leq x \leq 1/3$
- 2 - artylerzyści dla $1/3 < x \leq 2/3$
- 3 – konnica dla $2/3 < x \leq 1$

Natomiast 2., 4. i 6. gen oznaczają liczebność odpowiedniej jednostki według wzoru: $x * 1000 // 1$

Funkcja fitness

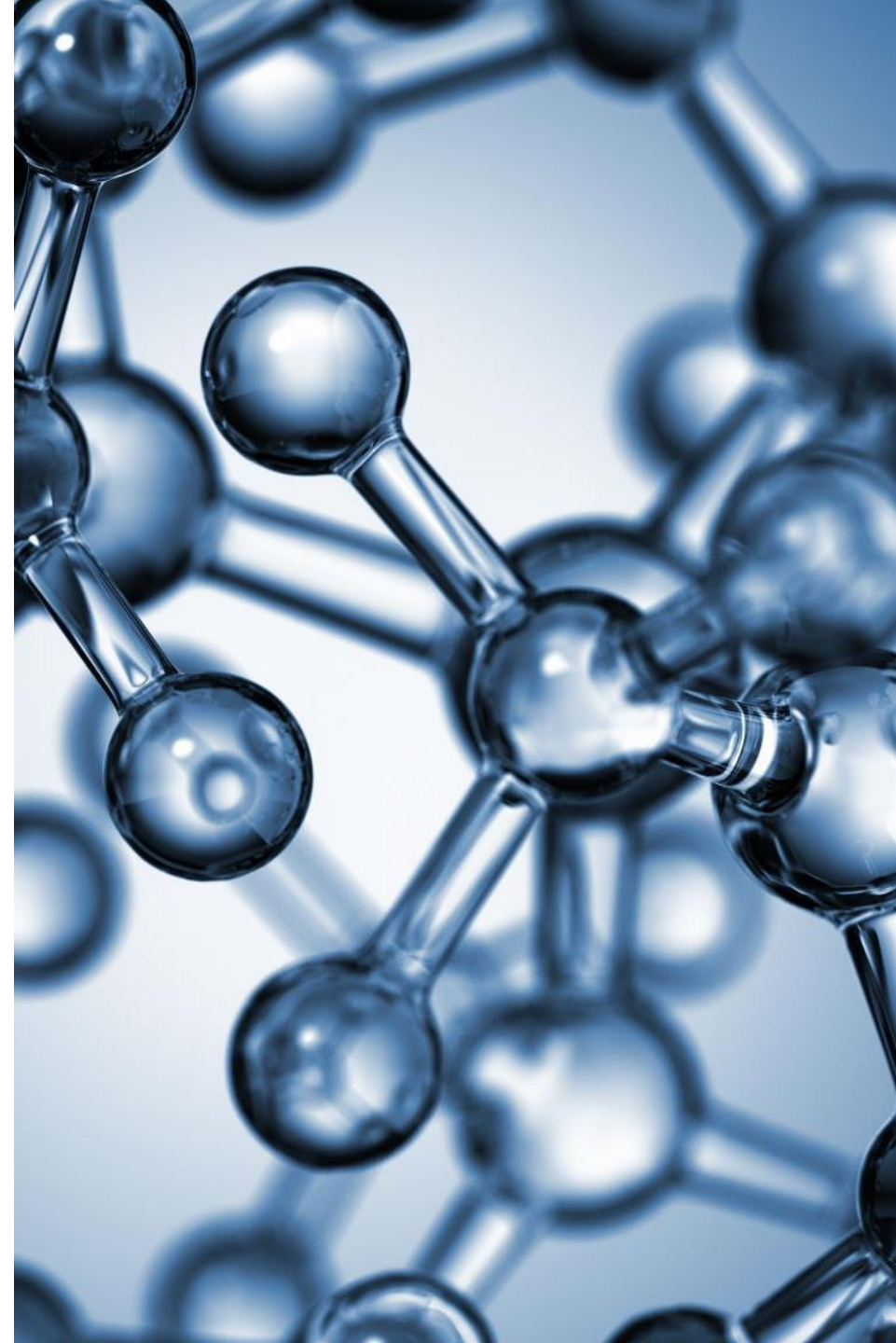
- Bardzo duża kara (100_000) za przekroczenie budżetu.
- 100 nagrody za wygranie walki na skrzydle, 200 kary. Kara większa, aby "zachęcić" do obstawienia obu skrzydeł.
- 500 nagrody lub kary za wynik bitwy w centrum (a więc całej bitwy) + pozostałe jednostki // 5 (dodatkowy impuls aby "zachęcić" do maksymalizacji ustawienia wojsk)



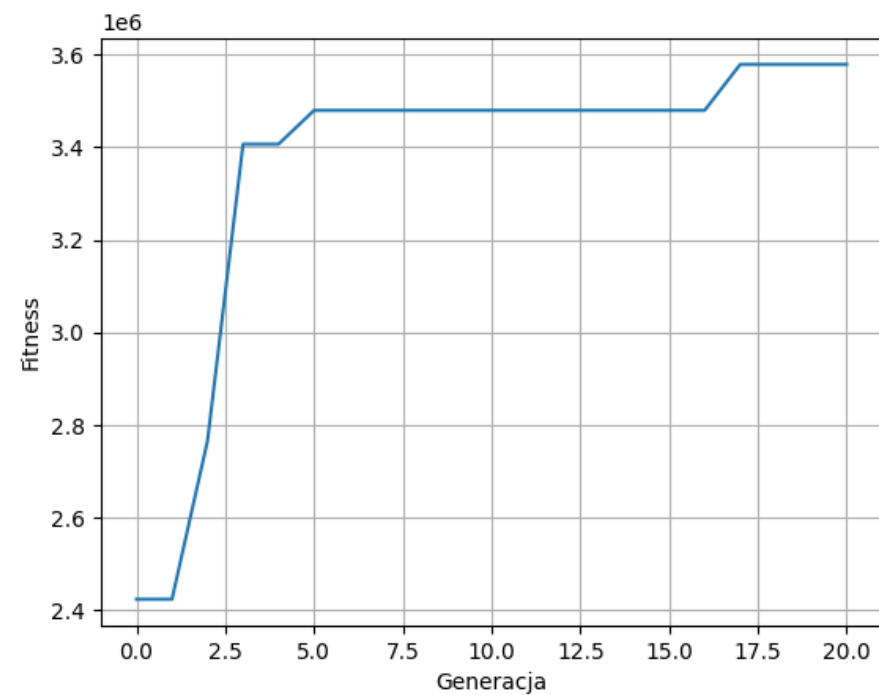
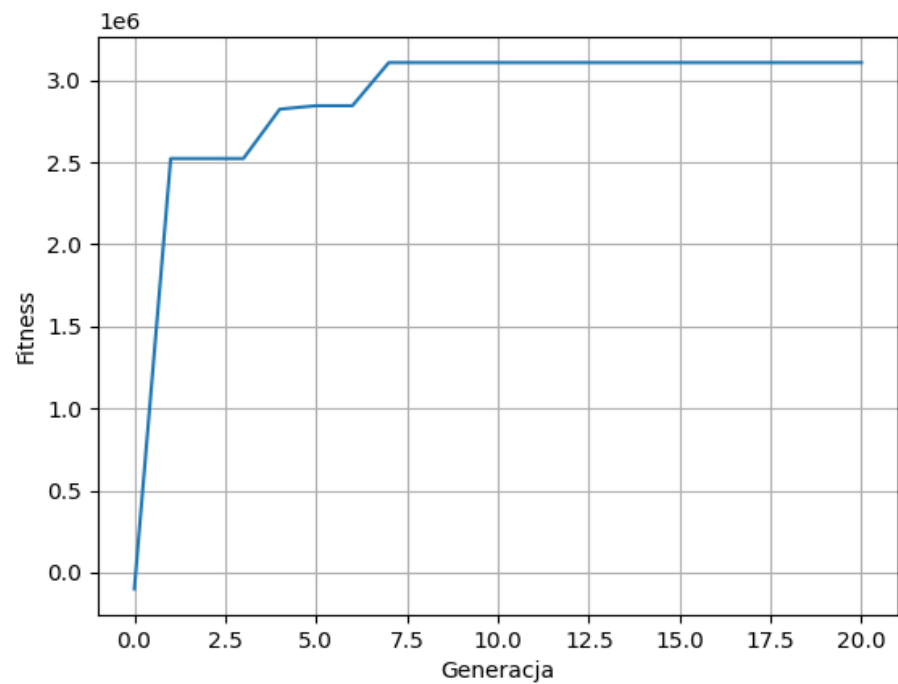
Krzyżowanie genów

Selekcja rodziców metodą "sss", z typem podziału "two-points"(ponieważ wojsko dzieli się na 3 części, tak jak w tym typie podziału).

20 chromosomów, 20 generacji, 18% mutacji, 8 najlepszych rodziców przekazuje geny, 2 rodziców przechodzi między generacjami (optymalne połączenie szybkości działania i wyników, testowałem różne współczynniki parametrów).



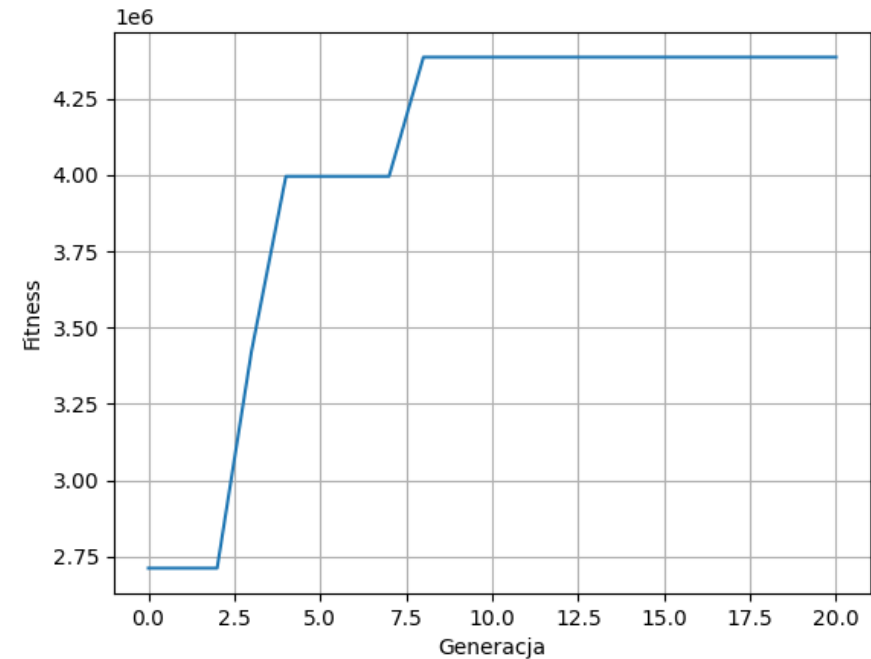
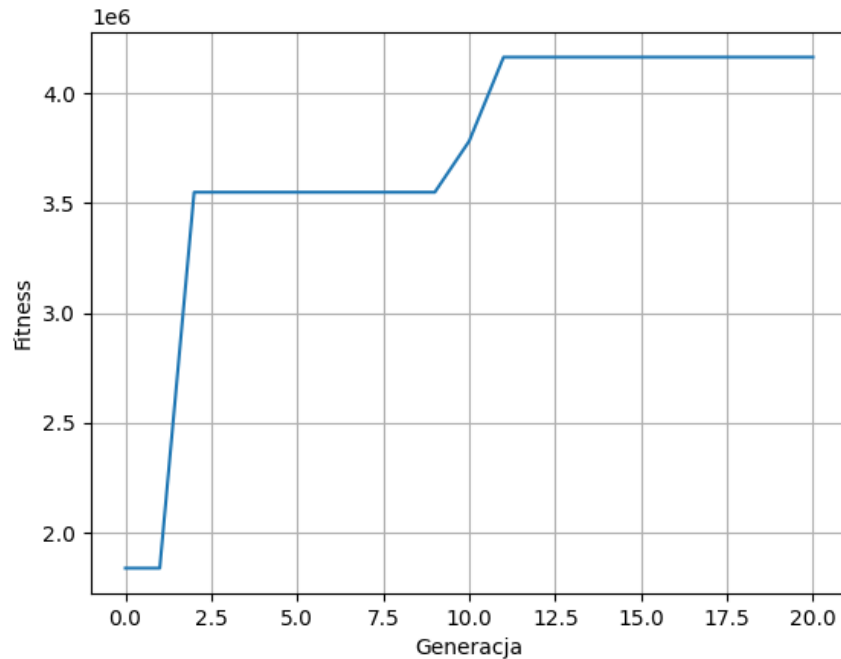
Wyniki



Wyniki

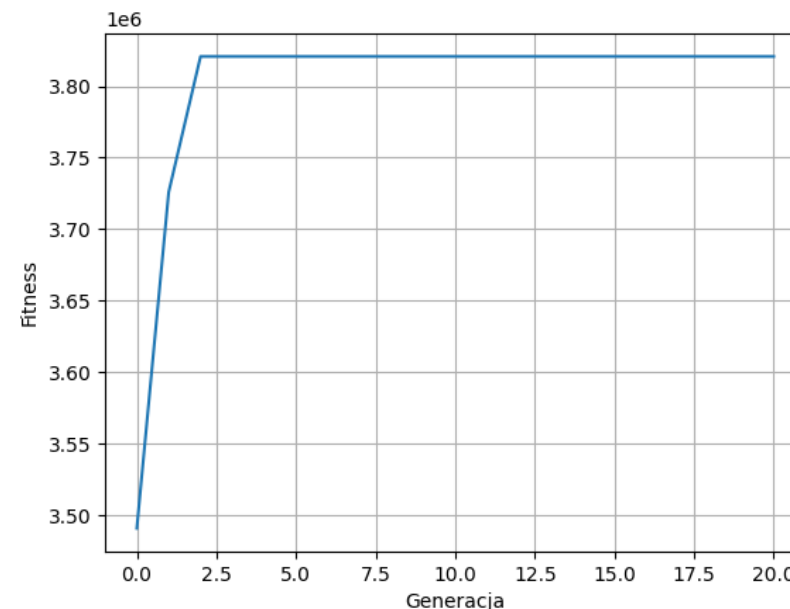
Dwa podejścia:

- Tylko liczba wygranych bitw
- Liczba wygranych bitw, ale wzięcie pod uwagę, jak często wygrywaliśmy na skrzydłach



Najlepsze rozwiązanie

Najlepszy wynik według drugiego podejścia wygląda następująco:



Parameters of the best solution : [0.34147156 0.37038776 1. 0. 0. 0.41857535]

Fitness value of the best solution = 3949277.0

Wykorzystany budżet: 973.0

Na 5000 bitw

Lewe skrzydło wygrało: 4617 92.34%

Prawe skrzydło wygrało: 4575 91.5%

Cała bitwa została wygrana: 4984 99.68%

Najlepsze rozwiązanie

Jak można zauważyć na poprzednim slajdzie, budżet nie został w pełni wykorzystany. Postanowiłem ręcznie zmodyfikować ten wynik tak, aby osiągnąć jeszcze lepsze rozwiązanie. Z kilku prób, w których w pełni wykorzystałem budżet, najlepszy wygląda następująco:

```
[0.5 0.406 1. 0. 0. 0.391]
```

```
Wykorzystany budżet: 1000.0
```

```
Na 5000 bitw
```

```
Lewe skrzydło wygrało: 4728 94.56%
```

```
Prawe skrzydło wygrało: 4505 90.1000000000000001%
```

```
Cała bitwa została wygrana: 5000 100.0%
```

Dziękuję za uwagę

Marcin Wojciechowski