

**Wydział Elektroniki i Technik Informacyjnych
Politechnika Warszawska**

Algorytmy Ewolucyjne

Sprawozdanie z projektu 2

Michał Kwarciński, Kacper Marchlewicz

Warszawa, 2023

Spis treści

1. Wstęp	2
1.1. Treść zadania	2
2. Prezentacja wyników	3
2.1. Dobór optymalnych parametrów algorytmu i metody selekcji	3
2.2. Osiągnięty optymalny wynik	6

1. Wstęp

1.1. Treść zadania

Stosując algorytm genetyczny znajdź rozwiązanie problemu plecakowego:

$$\max_x \sum_{i=1}^n p_i x_i \tag{1.1}$$

$$\sum_{i=1}^n w_i x_i \leq W \tag{1.2}$$

$$p_i > 0, w_i > 0, x \in \{0, 1\} \tag{1.3}$$

Założenia

- liczba przedmiotów: $n = 32$
- do generacji przedmiotów wykorzystać Skrypt 1; w przypadku wykonania projektu w parze należy wybrać niższy numer albumu; wagi w przedmiotów są losowane z rozkładem równomiernym z przedziału $(0.1, 1)$ z dokładnością do 0.1, a wartości p przedmiotów są losowane z rozkładem równomiernym z przedziału: $(1, 100)$ z dokładnością do 1
- maksymalna waga plecaka: $W = 30\%$ wagi wszystkich przedmiotów
- dozwolone jest korzystanie ze środowiska MATLAB wraz z dodatkiem Global Optimization Toolbox (optimtool). Wykonanie projektu w Pythonie wymaga uprzedniej konsultacji z prowadzącym projekt.

2. Prezentacja wyników

2.1. Dobór optymalnych parametrów algorytmu i metody selekcji

Rodzaj sukcesji i mutacji zdecydowaliśmy wybrać się domyślny dla Matlab. Sukcesja jest sukcesją elitarną to a mutacji to możliwa adaptacja mutacji.

Zgodnie z zaleceniami na wykładzie dobraliśmy następujące parametry:

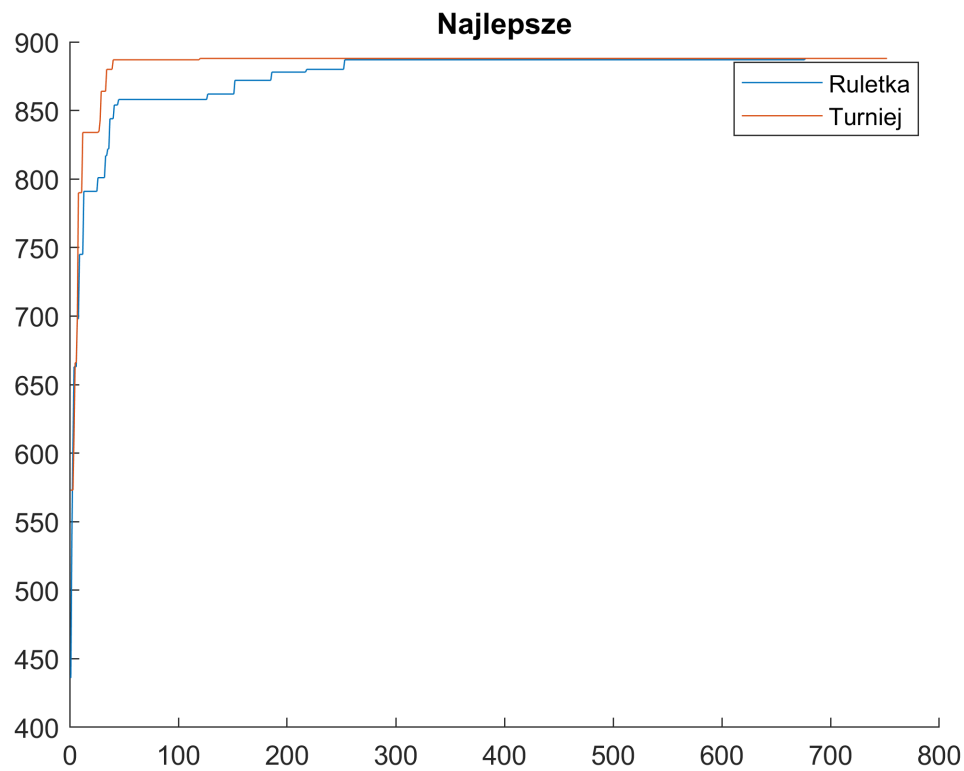
- licznosc elity równą 1
- stosunek osobników skrzyżowanych do nieskrzyżowanych na 80%
- prawdopodobieństwo mutacji na 20%
- brak migracji

Wynik przedstawiany jest jako wektor 32 liczb. Zdecydowaliśmy się na liczbę populacji równą 75 i maksymalną liczbę generacji 750. Pozwala to na osiągnięcie wyniku przy rozsądnym wykorzystaniu zasobów obliczeniowych. Postanowiliśmy dodać kolejny warunek stopu - 'MaxStallGenerations', odpowiedzialny za zatrzymanie algorytmu gdy średnie zmian dopasowania nowych generacji praktycznie nie będą się różnić, ustawiliśmy na 750, aby otrzymać lepsze porównanie algorytmów.

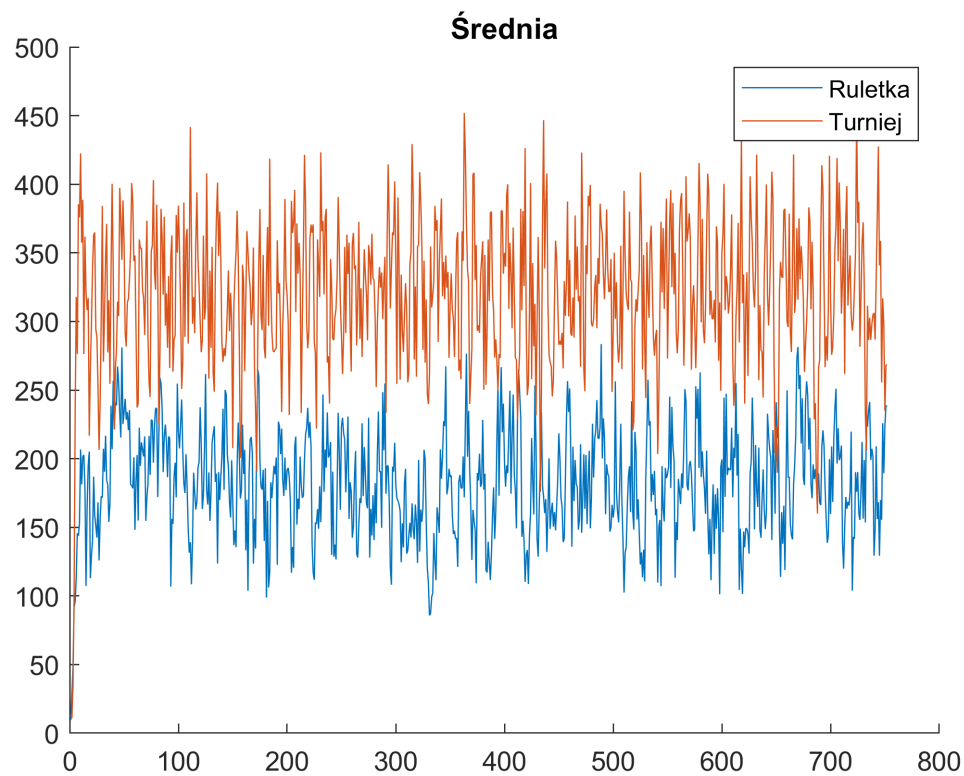
Następnie porównaliśmy ze sobą dwie metody selekcji - ruletkową i turniejową. Licznosc turniejowej jest domyślna, wynosi 4. Poniżej znajdują się wykresy następujących funkcji celu w funkcji numeru pokolenia:

- wartości maksymalnych 2.1
- średniej 2.2
- wariancji 2.3
- wartości minimalnych 2.4

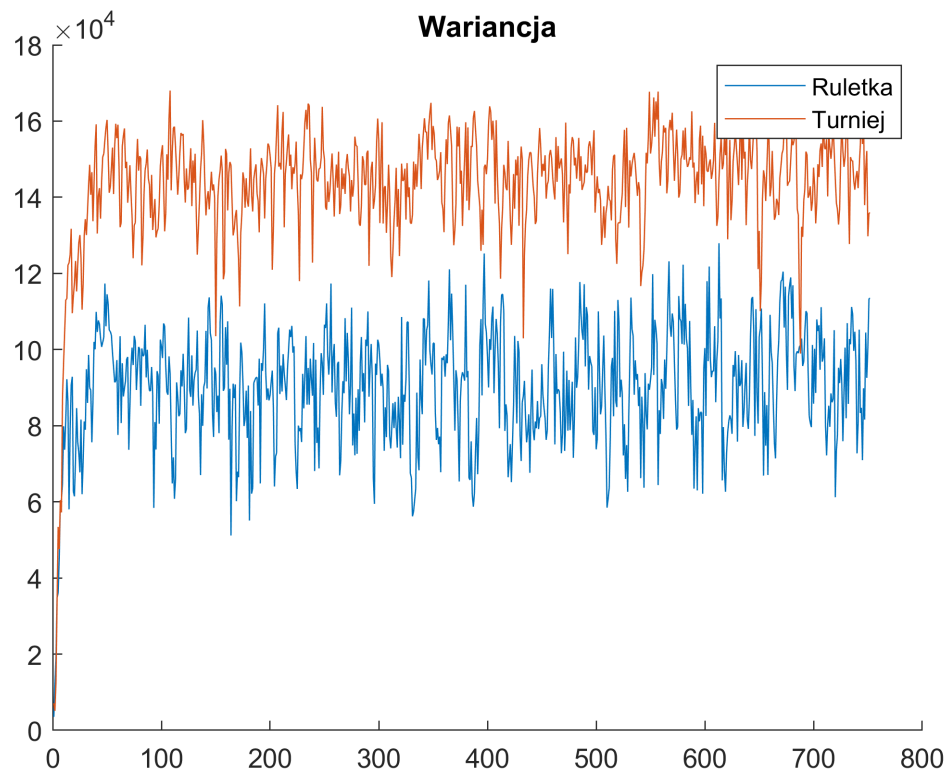
Możemy zauważyć, że selekcja turniejowa znacznie szybciej osiągnęła wartości zbliżone do optymalnego wyniku niż ruletkowa. Z uwagi na to widać również wyższą średnią i wariancję szukanego optimum. Oba algorytmy natomiast do końca działania algorytmu w populacji posiadały osobniki bardzo słabe - o wartościach funkcji celu równych zero. Mimo to można stwierdzić, że selekcja turniejowa sprawdza się lepiej, w populacji jest większa liczba porządkanych osobników, widać jej przewagę w zdolnościach eksploracyjnych.



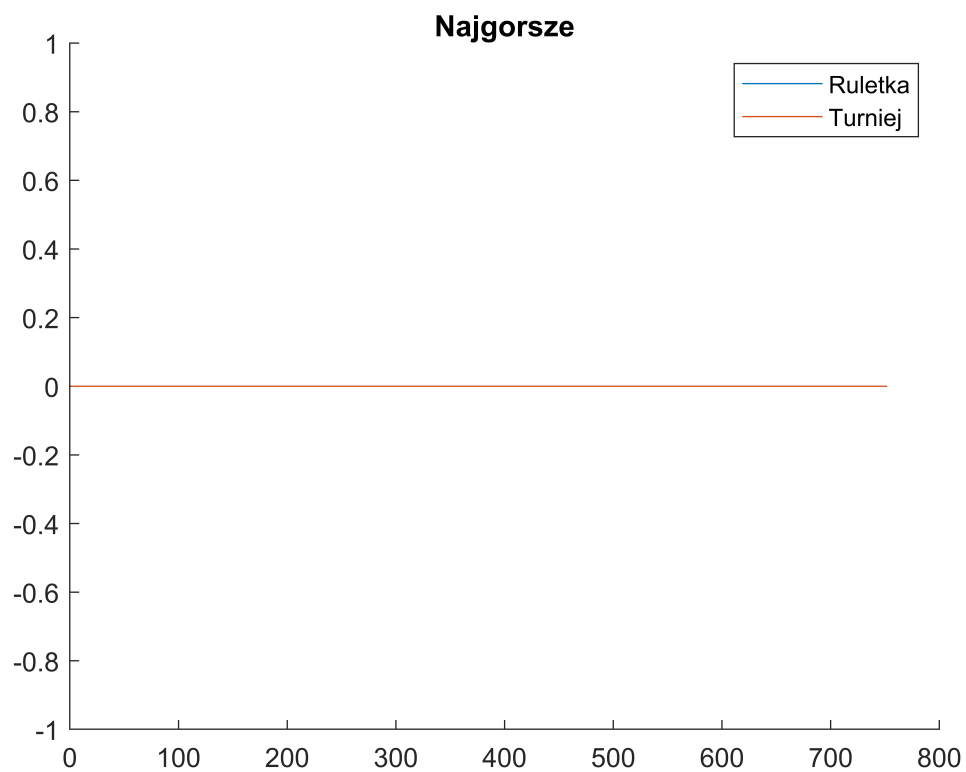
Rys. 2.1: Wykres najlepszej wartości funkcji celu w funkcji numeru pokolenia



Rys. 2.2: Wykres średniej wartości funkcji celu w funkcji numeru pokolenia



Rys. 2.3: Wykres wariacji wartości funkcji celu w funkcji numeru pokolenia



Rys. 2.4: Wykres najgorszej wartości funkcji celu w funkcji numeru pokolenia

2.2. Osiągnięty optymalny wynik

Maksymalna waga = 4,56

Poszukiwane rozwiązanie problemu plecakowego prezentuje się następująco:

wartość przedmiotu	waga przedmiotu	wektor rozwiązań
7	0,1	0
50	0,5	0
58	0,6	0
41	0,7	0
25	0,7	0
14	0,1	1
14	0,6	0
94	0,3	1
55	0,2	1
93	0,4	1
3	0,9	0
78	0,8	0
95	0,5	1
16	0,7	0
24	0,1	1
34	1	0
42	0,3	1
56	0,2	1
24	0,6	0
12	0,2	0
10	0,4	0
14	0,6	0
2	0,7	0
20	0,2	0
79	0,5	1
58	0,5	1
3	0,5	0
96	0,2	1
98	0,5	1
56	0,4	1
28	0,3	1
58	0,9	0

Oba warianty selekcji doszły do tego samego wektora rozwiązań. Wartość końcowa plecaka wynosi 888, a waga 4,5.