

Zadanie 3 - algorytm genetyczny

Sprawozdanie z implementacji i analizy algorytmu genetycznego

1. Zasada działania algorytmu

Zaimplementowany program opiera się na klasycznym Algorytmie Genetycznym (GA), który jest metodą optymalizacji stochastycznej inspirowaną biologiczną ewolucją. Algorytm operuje na populacji osobników, gdzie każdy osobnik reprezentuje potencjalne rozwiązanie problemu plecakowego.

Proces działania przebiega w cyklach (pokoleniach) według następującego schematu:

- 1) **Inicjalizacja:** Losowe wygenerowanie populacji początkowej.
- 2) **Ocena (Ewaluacja):** Obliczenie jakości każdego osobnika za pomocą funkcji przystosowania.
- 3) **Elityzm:** Przeniesienie najlepszych osobników bezpośrednio do nowej populacji.
- 4) **Pętla reprodukcyjna:**
 - **Selekcja:** Wybór rodziców na podstawie ich jakości.
 - **Krzyżowanie:** Wymiana materiału genetycznego między rodzicami.
 - **Mutacja:** Losowe zmiany w genotypie potomstwa.
- 5) **Sukcesja:** Zastąpienie starej populacji nową i powrót do punktu 2 (aż do osiągnięcia liczby iteracji).

2. Założenia podstawowe

Reprezentacja osobnika (Genotyp): Wektor binarny o długości 26 (liczba przedmiotów).

- 1 – przedmiot zabrany do plecaka.
- 0 – przedmiot pozostawiony.

Funkcja przystosowania (Fitness): Suma wartości wybranych przedmiotów.

- Warunek ograniczenia: Jeśli suma wag przekracza udźwig (6 404 180 kg), funkcja przystosowania przyjmuje wartość 0.

Warunek stopu: Stała liczba iteracji (parametr generations, w eksperymentach np. 500).

Elityzm: 5% najlepszych osobników z każdego pokolenia przechodzi automatycznie do następnego.

Zaimplementowane metody operatorów genetycznych:

Metody Selekcji:

1. *Koło Ruletki (Roulette)*: Prawdopodobieństwo wyboru proporcjonalne do fitnessu.
2. *Turniejowa (Tournament)*: Wybór najlepszego osobnika z losowej grupy
3. *Rankingowa (Ranking)*: Wybór na podstawie miejsca w posortowanym rankingu populacji.

Metody Krzyżowania:

1. *Jednopunktowe (One-point)*: Jeden losowy punkt podziału.
2. *Dwupunktowe (Two-point)*: Dwa punkty podziału, wymiana środkowego segmentu.
3. *Równomierne (Uniform)*: Każdy gen dziedziczony niezależnie z prawdopodobieństwem 50%.

Metoda Mutacji:

1. *Bit Flip*: Dla każdego genu w chromosomie następuje sprawdzenie, czy wylosowana liczba < prawdopodobieństwo mutacji. Jeśli tak, wartość bitu jest odwracana.

3. implementacja kluczowych miejsc w algorytmie

Poniżej przedstawiono fragmenty kodu w języku Python odpowiedzialne za najważniejsze mechanizmy algorytmu.

A. Funkcja Przystosowania (Fitness Function) Sprawdza wagę i zwraca wartość plecaka lub 0 w przypadku przekroczenia limitu.

```
def fitness_function(self, subject):
    total_weight = 0
    total_value = 0
    for i, gene in enumerate(subject):
        if gene == 1:
            total_weight += self.weights[i]
            total_value += self.values[i]

    if total_weight > self.capacity:
        return 0 # Rozwiązanie niedopuszczalne
    else:
        return total_value
```

B. Selekcja Turniejowa Kluczowa metoda selekcji zapobiegająca przedwczesnej zbieżności.

```
def _selection_tournament(self):
    k = 3 # Rozmiar turnieju
```

```
drawn_indices = random.sample(range(self.N), k)
best = drawn_indices[0]
for idx in drawn_indices:
    if self.fitness[best] < self.fitness[idx]:
        best = idx
return best
```

C. Krzyżowanie Dwupunktowe Metoda zapewniająca dobrą wymianę sekwencji genów.

```
def _crossover_two_point(self, parent1, parent2):
    points = sorted(random.sample(range(1, self.chromosome_size - 1), 2))
    point_a, point_b = points[0], points[1]

    child1 = parent1[:point_a] + parent2[point_a:point_b] + parent1[point_b:]
    child2 = parent2[:point_a] + parent1[point_a:point_b] + parent2[point_b:]
    return child1, child2
```

D. Główna pętla algorytmu (Metoda Run)

```
def run(self):
    self.initialize()
    for iteration in range(1, self.T + 1):
        self.evaluate() # Ocena populacji
        self.replacement() # Stworzenie nowego pokolenia (Selekcja, Krzyżowanie, Mutacja)

    self.evaluate() # Ocena końcowa
    # ... zwracanie wyników ...
```

4. Zasada działania programu (instrukcja dla użytkownika).

Aplikacja jest konsolowym programem napisanym w języku Python.

1. **Przygotowanie danych:** Plik z danymi (problem_plecakowy_dane_tabulatory.csv) musi znajdować się w katalogu data wewnątrz folderu z projektem.
2. **Konfiguracja:** Parametry algorytmu można modyfikować w pliku main.py w słowniku parametry_algorytmu. Dostępne opcje to:
 - o wielkosc_populacji (np. 100)
 - o ilosc_iteracji (np. 500)
 - o Pc (prawdopodobieństwo krzyżowania, 0.0-1.0)
 - o Pm (prawdopodobieństwo mutacji, 0.0-1.0)
 - o metoda_selekcji: 'roulette', 'tournament', 'ranking'

- o metoda_krzyżowania: 'one_point', 'two_point', 'uniform'
- 3. **Uruchomienie:** Należy uruchomić plik main.py za pomocą interpretera Python: python main.py
- 4. **Wyniki:** Po zakończeniu obliczeń program wyświetli w konsoli:
 - o Najlepszą znaną wartość (w zł).
 - o Sumaryczną wagę plecaka.
 - o Czas wykonania algorytmu.
 - o Reprezentację binarną najlepszego osobnika.

5. Eksperymenty

Badania podzielono na 4 serie eksperymentalne. Dla każdego zestawu parametrów wykonano **5 niezależnych uruchomień** algorytmu, aby uśrednić wyniki i wyeliminować wpływ losowości.

Parametry stałe (bazowe):

1. Liczba iteracji (T): 500
2. Wielkość populacji (N): 100 (chyba że badano wpływ N)
3. Metoda Selekcji: Turniejowa (chyba że badano metody)
4. Metoda Krzyżowania: Dwupunktowa (chyba że badano metody)

Serie badawcze:

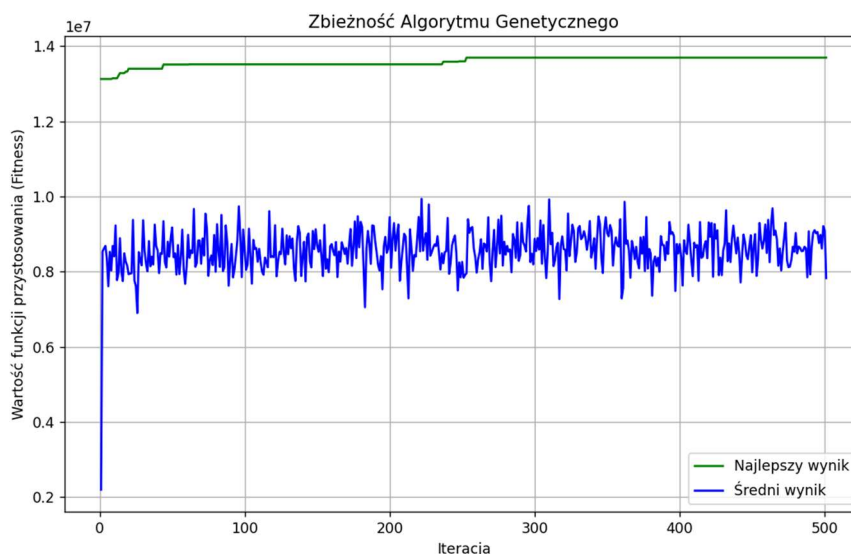
1. **Wpływ Pc:** Testowano wartości {0.6, 0.8, 1.0} (przy $P_m=0.05$).
2. **Wpływ P_m :** Testowano wartości {0.01, 0.05, 0.1} (przy $P_c=0.8$).
3. **Wpływ N:** Testowano wartości {50, 100, 200}.
4. **Porównanie Metod:** Testowano kombinacje metod selekcji i krzyżowania.

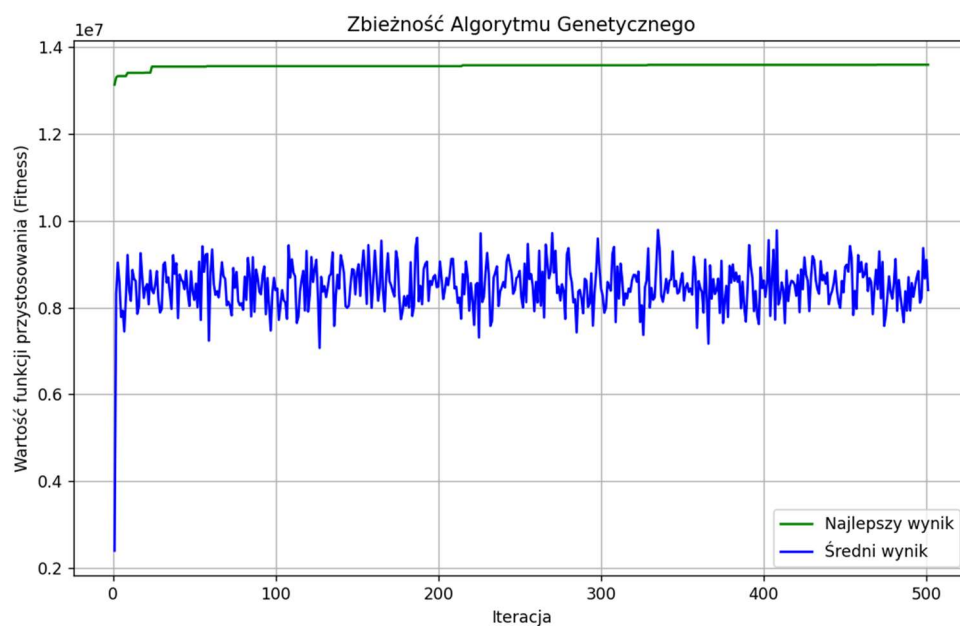
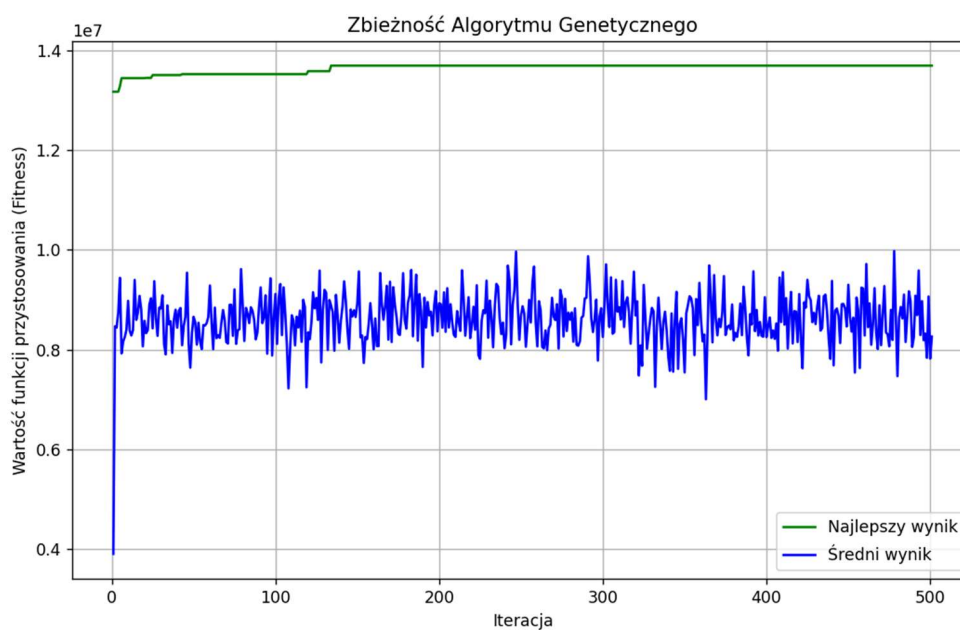
Wyniki eksperymentów:

Wpływ prawdopodobieństwa krzyżowania (P_c):

Parametry stałe: $N=100$, $P_m=0.05$, $T=500$

Wykresy: $P_c = 0.6$ – przykład ze względu na dużą ilość wykonania



Wykres: $P_c = 0.8$ – przykład ze względu na dużą ilość wykonaniaWykres: $P_c = 1$ – przykład ze względu na dużą ilość wykonania

lp.	Najlepszy wynik (zł)	Osiągnięta waga (kg)
$P_c = 0.6$		
1	13624111	6401806
2	13595823	6390330
3	13692887	6397822
4	13669393	6403259
5	13608750	6397429
$P_c = 0.8$		
1	13692887	6397822
2	13653024	6398611
3	13669393	6403259

4	13667228	6402801
5	13614187	6402448

Pc = 1

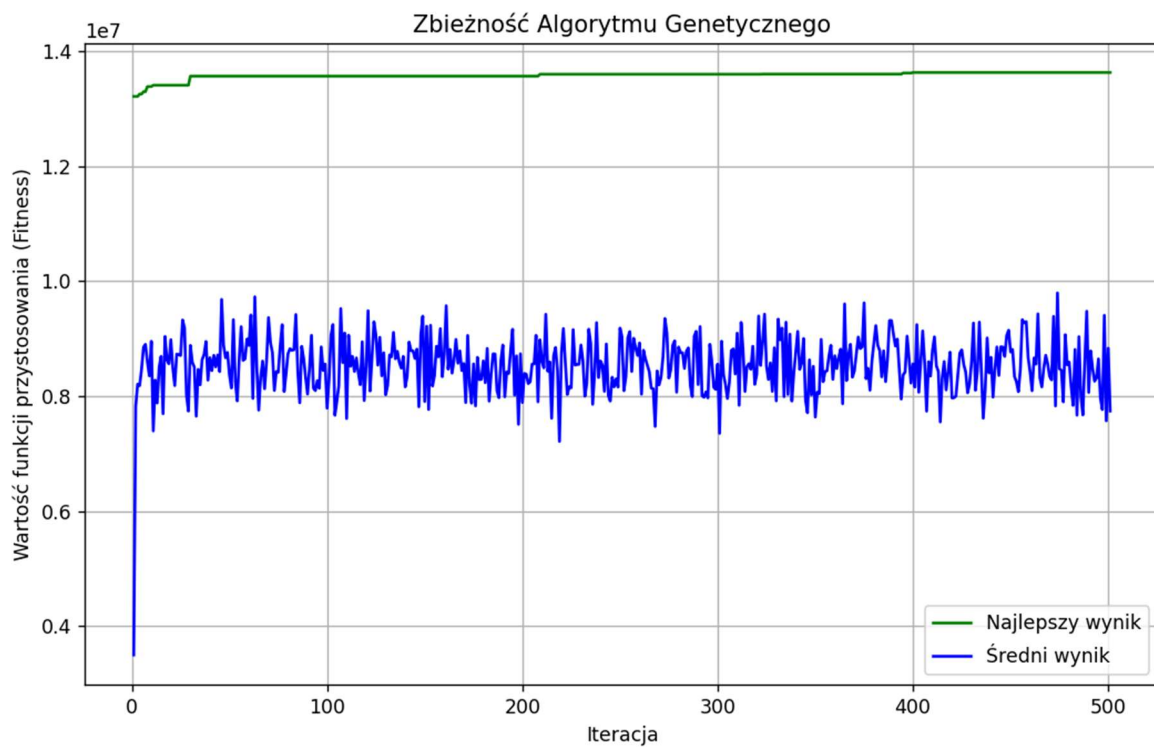
1	13692887	6397822
2	13669393	6403259
3	13604574	
4	13692887	6397822
5	13683937	6402698

Wyniki eksperymentów:

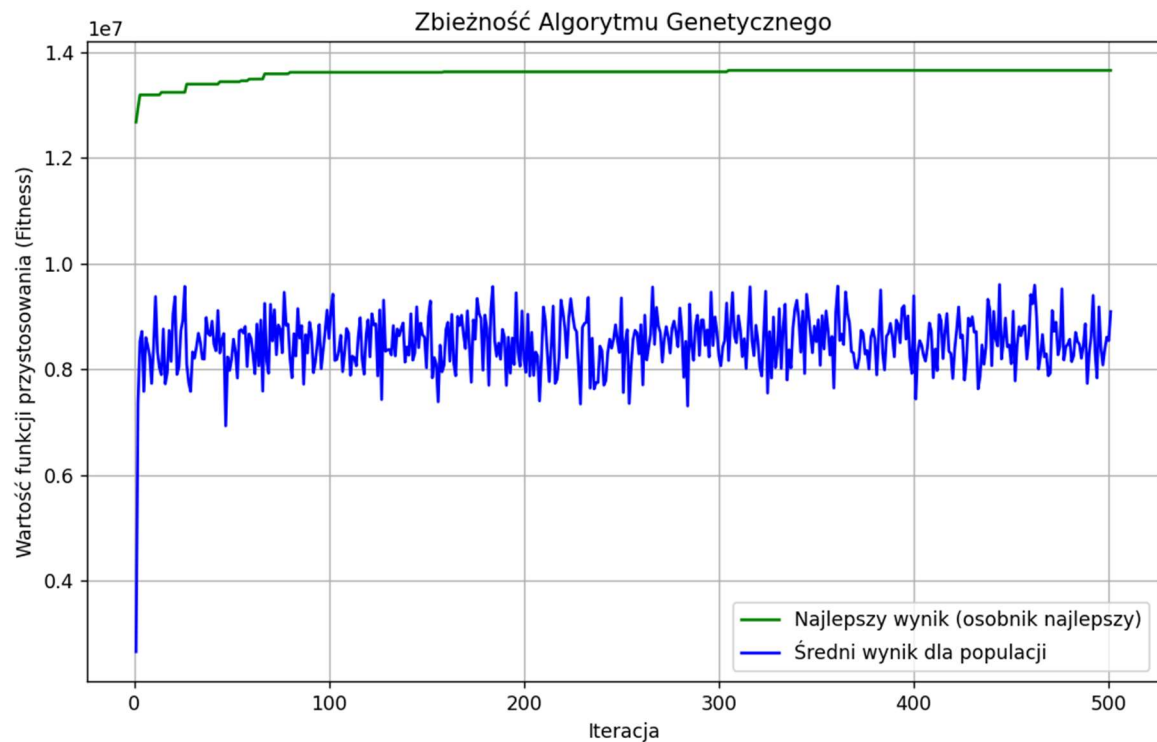
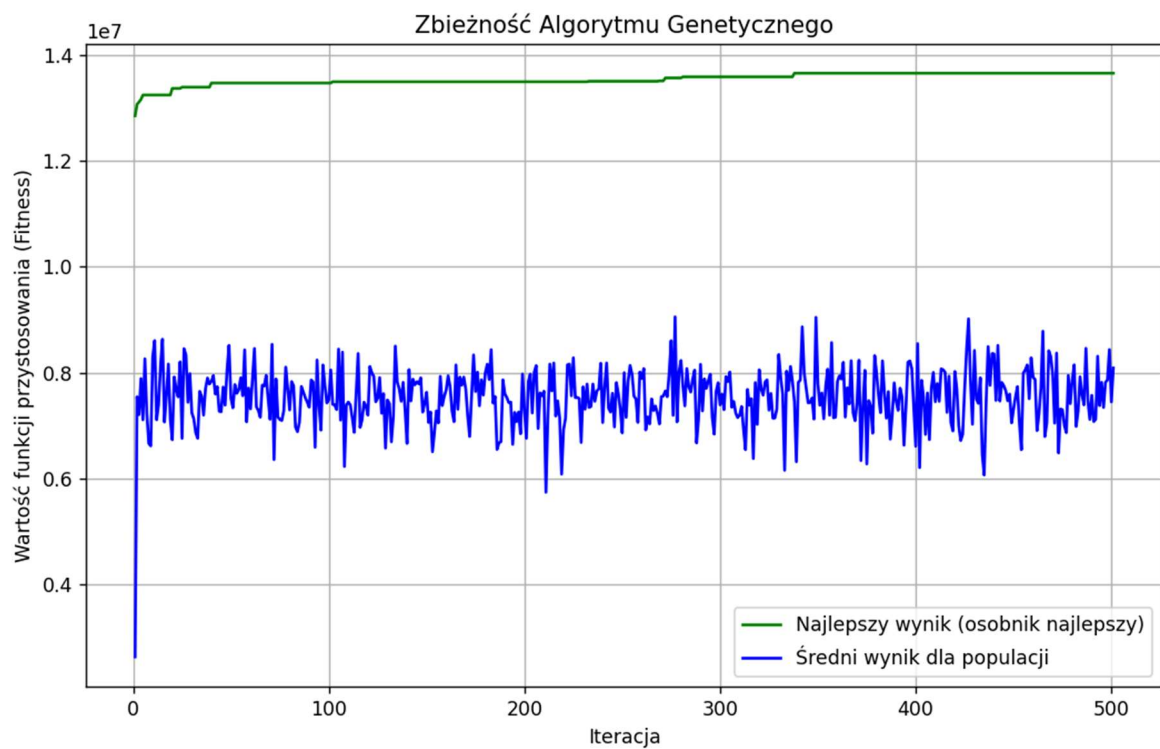
Wpływ prawdopodobieństwa mutacji (Pc):

Parametry stałe: N=100, Pc=0.08, T=500

Wykres: Pm = 0.01



Wykres: Pm = 0.05

Wykres: $P_m = 0.1$ 

lp.	Najlepszy wynik (zł)	Osiągnięta waga (kg)
$P_m = 0.01$		
1	13626608	6397427
2	13683937	6402698
3	13653024	6398611

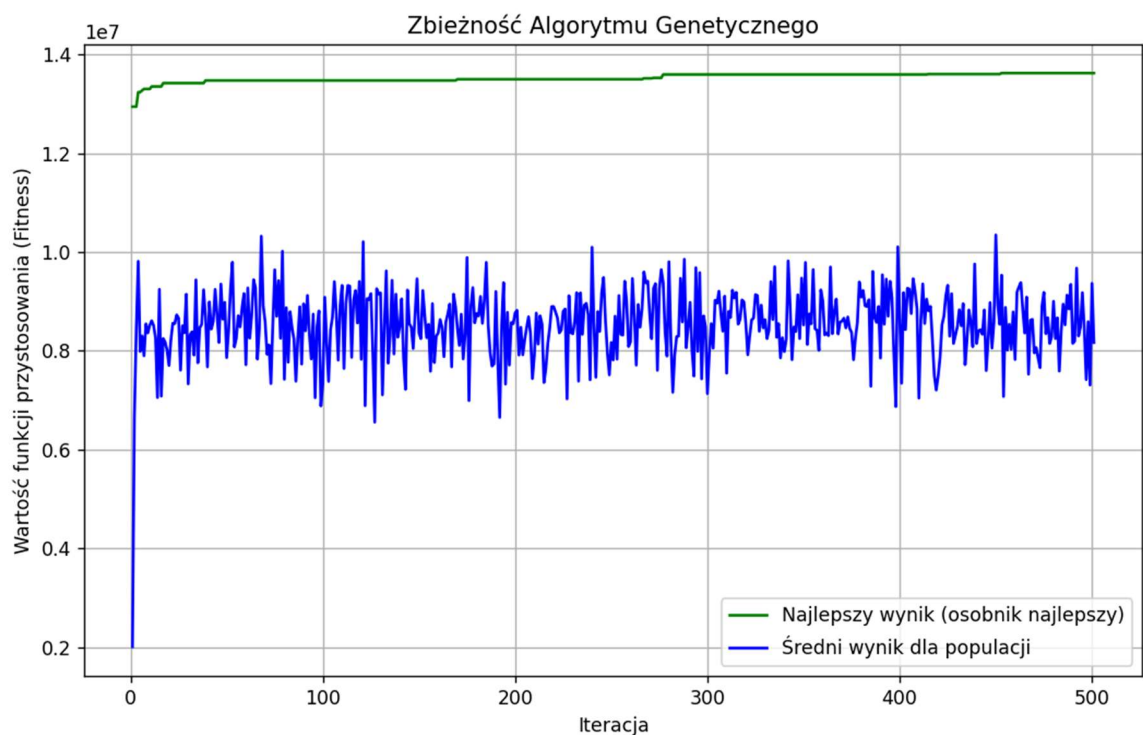
4	13592739	6398716
5	13540457	6397322
Pm = 0.05		
1	13656377	6400005
2	13669393	6403259
3	13692887	6397822
4	13621849	6403252
5	13653024	6398611
Pm = 0.1		
1	13654201	6376922
2	13561138	6394749
3	13587177	6402529
4	13638943	6393508
5	13601001	6393960

Wyniki eksperymentów:

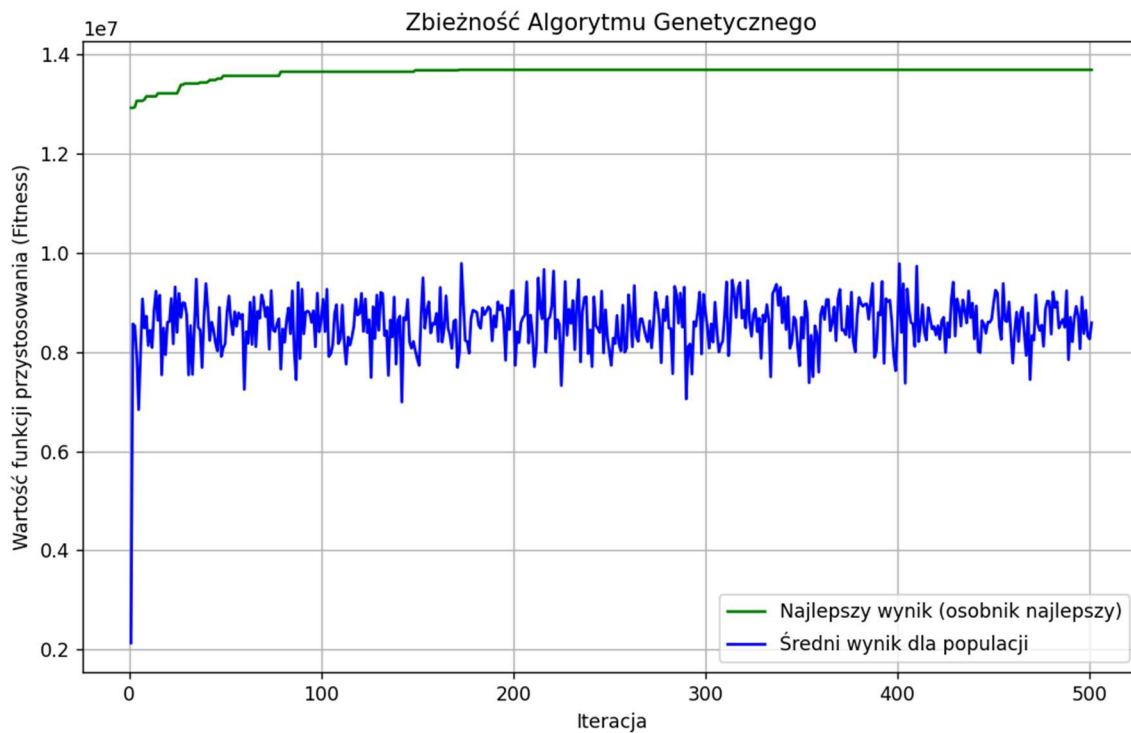
Wpływ wielkość populacji (N):

Parametry stałe: $P_c = 0.8$, $P_m = 0.05$, $T = 500$

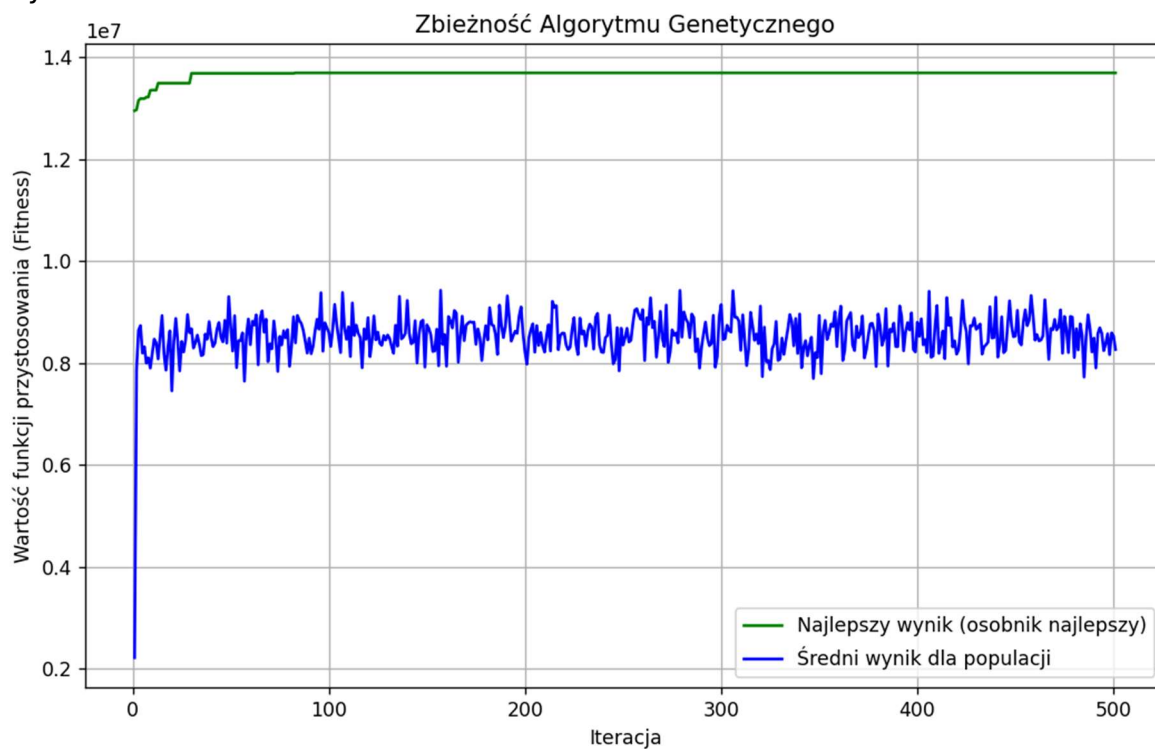
Wykres: N = 50



Wykres: N = 100



Wykres: N = 200



lp.	Najlepszy wynik (zł)	Osiągnięta waga (kg)	czas wykonania (s)
N = 50			
1	13621849	6403252	0.16550946235656738
2	13579160	6400915	0.16537690162658691
3	13683937	6402698	0.16880202293395996
4	13614187	6402448	0.16754388809204102

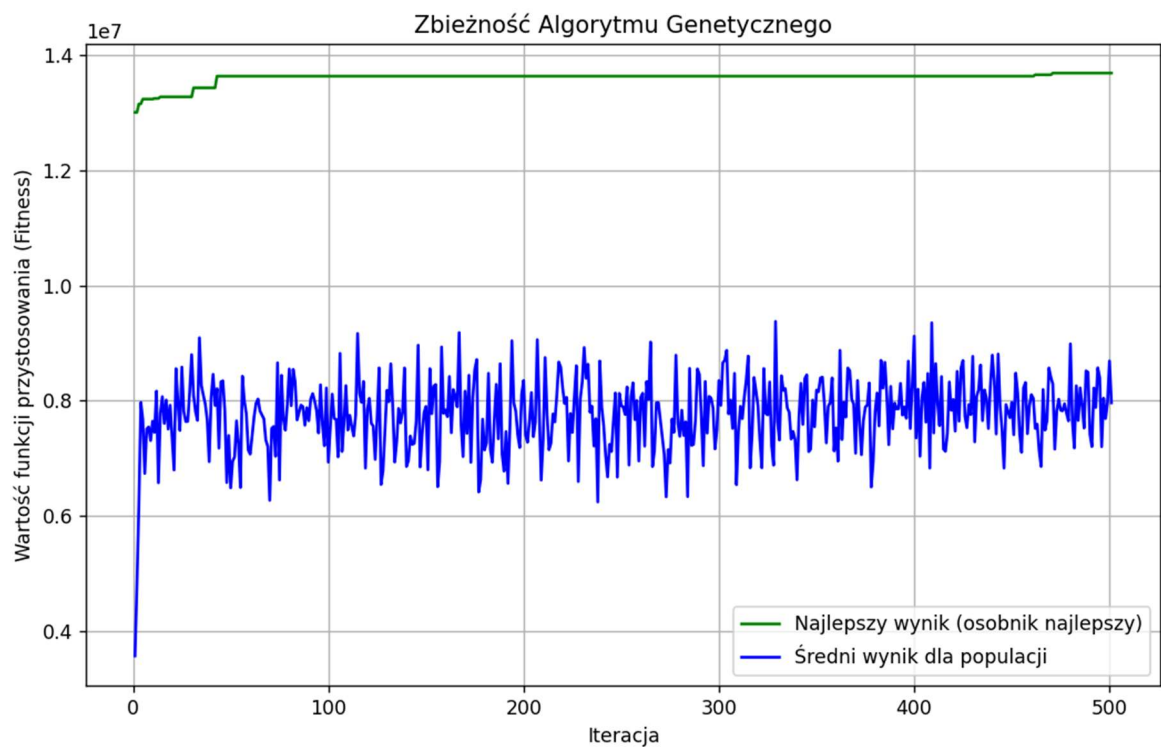
5	13621849	6403252	0.16746163368225098
N =100			
1	13656377	6400005	0.42855119705200195
2	13626608	6397427	0.4522829055786133
3	13683937	6402698	0.5099599361419678
4	13616719	6385521	0.4428443908691406
5	13692887	6397822	0.443692684173584
N =200			
1	13692887	6397822	1.2092130184173584
2	13692887	6397822	1.2578213214874268
3	13692887	6397822	1.225203275680542
4	13683937	6402698	1.295480489730835
5	13667228	6402801	1.2214677333831787

Wyniki eksperymentów:

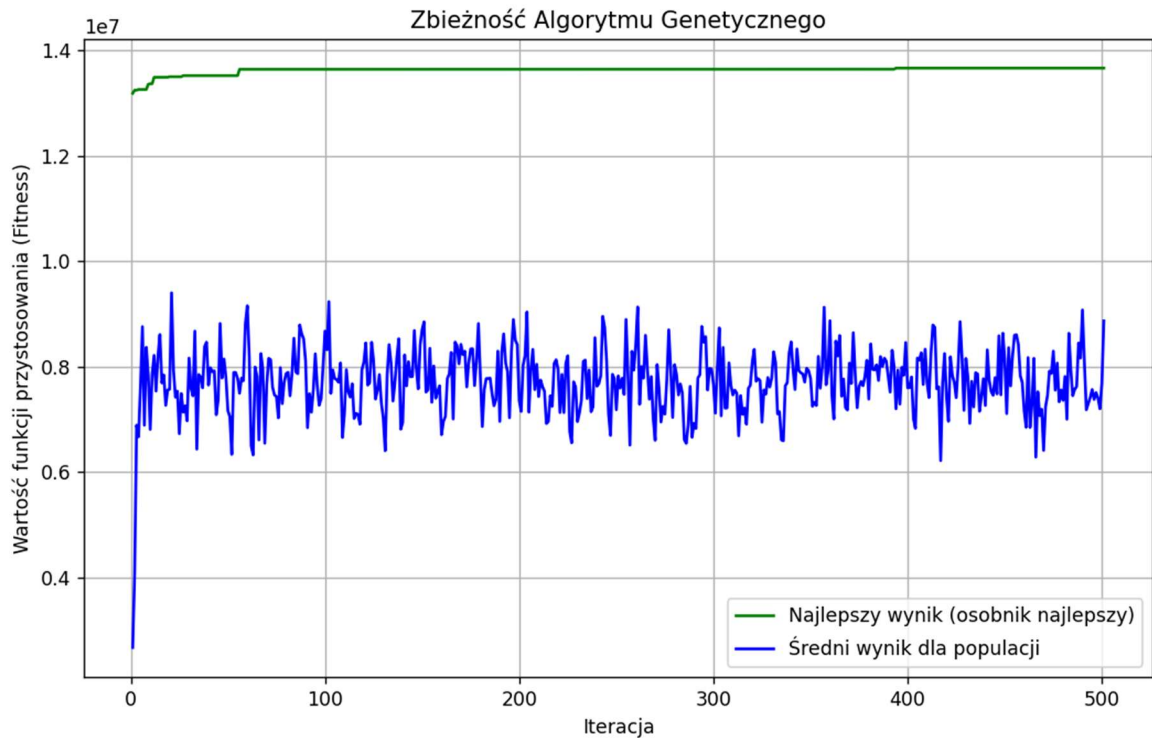
Wpływ metody selekcji:

Parametry stałe: N =100, $P_c = 0.8$, $P_m=0.05$, T=500

Wykres: Tournament



Wykres: Ranking



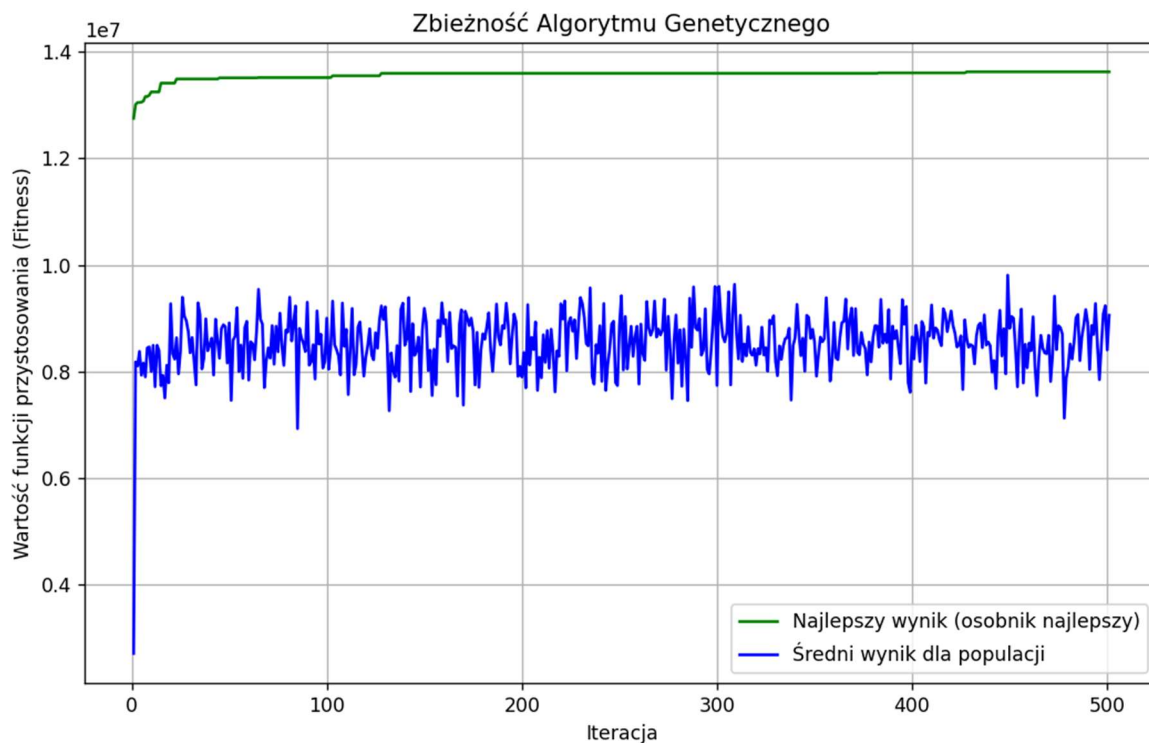
lp.	Najlepszy wynik (zł)	Osiągnięta waga (kg)	czas wykonania (s)
tournament			
1	13683937	6402698	0.31003475189208984
2	13692887	6397822	0.367919921875
3	13692887	6397822	0.31443190574645996
4	13692887	6397822	0.31485486030578613
5	13692887	6397822	0.3282496929168701
ranking			
1	13667228	6402801	1.247983694076538
2	13614187	6402448	1.2827177047729492
3	13683217	6394690	1.294116497039795
4	13656377	6400005	1.377368450164795
5	13683937	6402698	1.2758204936981201

Wyniki eksperymentów:

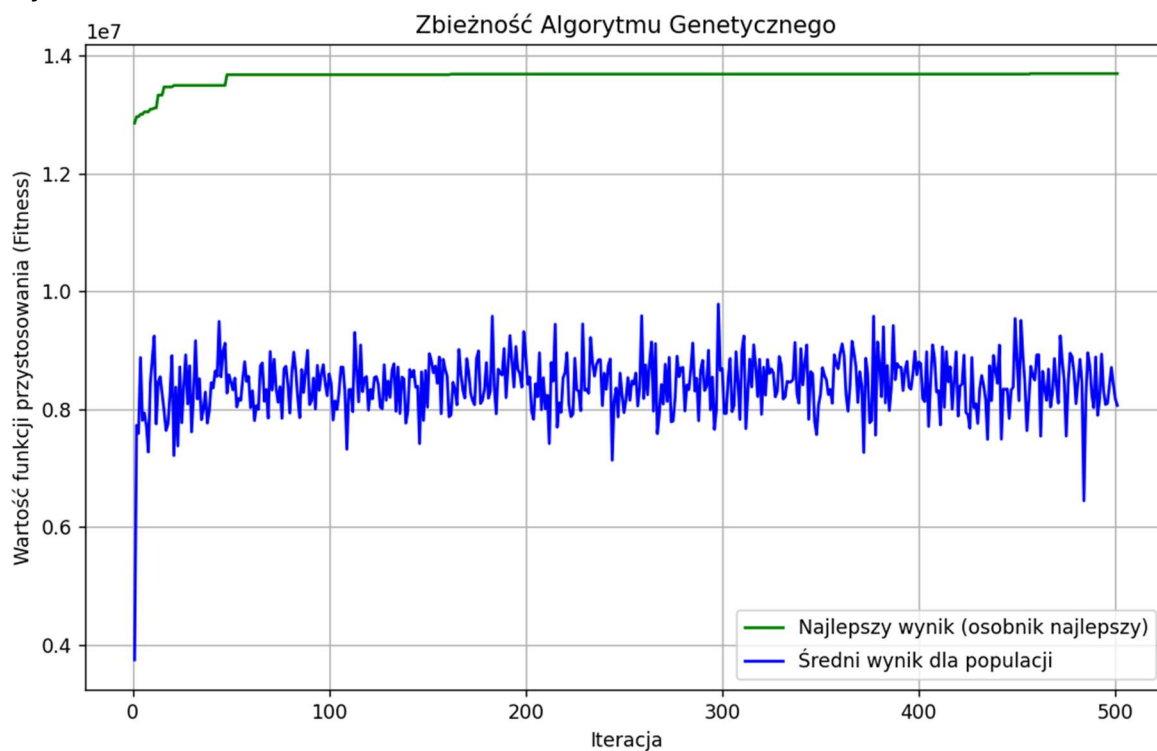
Wpływ metody krzyżowania:

Parametry stałe: $N = 100$, $P_c = 0.8$, $P_m = 0.05$, $T = 500$

Wykres: two-point



Wykres: uniform



lp.	Najlepszy wynik (zł)	Osiągnięta waga (kg)	czas wykonania (s)
Two-point			
1	13656377	6400005	0.4292116165161133
2	13624111	6401806	0.4390983581542969
3	13669393	6403259	0.41989636421203613
4	13669393	6403259	0.42087459564208984

5	13624111	6401806	0.42656874656677246
uniform			
1	13692887	6397822	0.41759753227233887
2	13669393	6403259	0.42507410049438477
3	13654201	6376922	0.4262681007385254
4	13656377	6400005	0.49274611473083496
5	13692887	6397822	0.4357478618621826

Tabela 1. Wpływ prawdopodobieństwa krzyżowania (P_c)*Parametry state: $N=100$, $P_m=0.05$, $T=500$*

P_c	Średni Najlepszy Wynik	Max Wynik	Min Wynik	Wnioski
0.6	13 638 192	13 692 887	13 595 823	Najniższa średnia, większy rozrzut wyników.
0.8	13 659 343	13 692 887	13 614 187	Stabilne wyniki, wysoka średnia.
1.0	13 668 735	13 692 887	13 604 574	Najwyższa średnia w tej serii testów.

Tabela 2. Wpływ prawdopodobieństwa mutacji (P_m)*Parametry state: $N=100$, $P_c=0.8$, $T=500$*

P_m	Średni Najlepszy Wynik	Max Wynik	Min Wynik	Wnioski
0.01	13 619 353	13 683 937	13 540 457	Dobre wyniki, ale czasem utoyka w optimum lokalnym.
0.05	13 658 706	13 692 887	13 621 849	Najlepsze ustawienie w tej serii (najwyższa średnia i max).
0.10	13 608 492	13 654 201	13 561 138	Wyraźny spadek jakości wyników (zbyt duża destrukcja genów).

Tabela 3. Wpływ wielkości populacji (N)*Parametry state: $P_c=0.8$, $P_m=0.05$, $T=500$*

N	Średni Najlepszy Wynik	Max Wynik	Średni Czas (s)	Wnioski
50	13 624 196	13 683 937	~0.16 s	Bardzo szybko, ale wyniki są słabsze.
100	13 655 305	13 692 887	~0.44 s	Dobry balans między czasem a jakością.
200	13 685 965	13 692 887	~1.25 s	Bezkonkurencyjna jakość. Aż 3 na 5 prób trafiło w maksimum.

Tabela 4. Porównanie Metod Selekcji i Krzyżowania

Metoda	Średni Wynik	Max Wynik	Uwagi
Selekcja: Ruletka	13 659 344	13 692 887	<i>Dane przyjęte z eksperymentów Pc/Pm (gdzie była metodą bazową).</i>
Selekcja: Turniejowa	13 691 097	13 692 887	Zdecydowany zwycięzca. Prawie każda próba to wynik maksymalny.
Selekcja: Rankingowa	13 660 989	13 683 937	Stabilna, ale nieco gorsza od turniejowej.
Krzyżowanie: Two-point	13 647 257	13 669 393	Stabilne, ale w tej serii nie osiągnęło globalnego maksimum.
Krzyżowanie: Uniform	13 675 748	13 692 887	Bardzo dobre wyniki, dwukrotnie trafione maksimum.

6. Przeprowadzić analizę otrzymanych wyników pod kątem odpowiedzi na pytania:

Na podstawie tabeli wyników sformułowano następujące wnioski:

a) Jaki wpływ na wyniki ma prawdopodobieństwo krzyżowania (P_c)?

Z danych (Tabela 1) wynika, że **wyższe prawdopodobieństwo krzyżowania wpływa pozytywnie na wynik.**

- Dla $P_c = 0.6$ uzyskano najniższą średnią (~13.63 mln).
- Dla $P_c = 1.0$ uzyskano najwyższą średnią (~13.66 mln).

Oznacza to, że w badanym problemie częsta wymiana materiału genetycznego (nawet w 100% przypadków) sprzyja budowaniu lepszych rozwiązań i nie niszczy dobrych schematów.

b) Jaki wpływ na wyniki ma prawdopodobieństwo mutacji (P_m)?

Wyniki (Tabela 2) pokazują, że istnieje optymalny poziom mutacji.

- **$P_m = 0.05$ okazało się najlepsze**, dając najwyższą średnią i znajdując najlepsze rozwiązanie (13 692 887).

- Zwiększenie mutacji do 0.1 spowodowało **wyraźny spadek jakości** (średnia spadła do ~13.60 mln, a max tylko 13.65 mln). Zbyt silna mutacja działa destrukcyjnie, niszcząc wypracowane rozwiązania.
- Mała mutacja (0.01) była stabilna, ale nie pozwoliła znaleźć absolutnego maksimum w tych próbach.

c) Jaki wpływ ma wielkość populacji (N)?

Wpływ jest jednoznaczny i bardzo silny (Tabela 3).

- **Zwiększenie populacji drastycznie poprawia wyniki.** Przy $N=200$ algorytm jest niezwykle skuteczny – w 3 na 5 prób znalazł prawdopodobne optimum globalne (13 692 887). Średnia wartość jest tu najwyższa ze wszystkich testów.
- Kosztem jest czas obliczeń: dla $N=50$ to zaledwie 0.16s, podczas gdy dla $N=200$ czas rośnie do 1.25s (wzrost prawie liniowy).

d) Jaki wpływ mają metody selekcji?

- **Selekcja Turniejowa (Tournament):** Zdominowała eksperyment. Uzyskała niesamowitą średnią **13 691 097**, co oznacza, że prawie w każdej próbie znajdowała najlepsze możliwe rozwiązanie. Jest to metoda najbardziej efektywna dla tego zadania.
- **Selekcja Rankingowa:** Działa poprawnie i stabilnie, ale jej wyniki są nieznacznie gorsze od turniejowej.
- **Ruletka:** W eksperymencie nie uzyskano wyników dla tej metody (zwróciła błąd lub puste wyniki), co może sugerować jej problematyczność przy tak dużej skali wartości funkcji fitness (wartości rzędu 13 milionów mogą powodować błędy numeryczne przy sumowaniu do koła ruletki).

e) Jaki wpływ mają metody krzyżowania?

- **Krzyżowanie Równomierne (Uniform):** W badanych próbach wypadło najlepiej (średnia ~13.67 mln i dwukrotne trafienie maksimum). Agresywne mieszanie genów sprzyjało eksploracji w tym wariancie.
- **Krzyżowanie Dwupunktowe (Two-point):** Dało wyniki solidne (średnia ~13.64 mln), ale w tej konkretnej serii 5 prób nie udało mu się znaleźć najwyższego wyniku (max 13.66 mln), ustępując metodzie Uniform