

Algorytmy Ewolucyjne - Projekt 2

Problem Plecakowy

Antoni Matczuk - 331129

Jakub Szubzda - 331145

25 maja 2025

Spis treści

1	Wstęp	3
2	Definicja Problemu	3
2.1	Generowanie Przedmiotów	3
2.2	Lista Wszystkich Przedmiotów	3
3	Algorytm Genetyczny	4
3.1	Funkcja Celu	4
3.2	Ograniczenia	5
3.3	Parametry Algorytmu Genetycznego	5
3.4	Kryteria Zatrzymania Algorytmu	5
3.5	Konfiguracja Algorytmu w MATLAB	5
4	Wyniki	6
4.1	Rozwiązanie Problemu	6
4.2	Wybrane Przedmioty	6
4.3	Statystyki Wykonania Algorytmu	6
4.4	Liczności Potomków w Każdej Generacji	7
4.5	Wykresy Przebiegu Algorytmu	7
5	Wnioski	9

1 Wstęp

Celem projektu jest znalezienie rozwiązania problemu plecakowego przy użyciu algorytmu genetycznego. Problem plecakowy polega na wyborze takiego podzbioru przedmiotów spośród danego zbioru, aby suma wartości wybranych przedmiotów była jak największa, nie przekraczając jednocześnie określonego limitu wagowego plecaka. W ramach projektu wykorzystano środowisko MATLAB wraz z Global Optimization Toolbox.

2 Definicja Problemu

Problem plecakowy jest klasycznym problemem optymalizacyjnym. W wersji 0/1, dla każdego przedmiotu podejmujemy decyzję, czy zabrać go w całości, czy też nie. Matematycznie problem można sformułować następująco:

Maksymalizuj:

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

Przy ograniczeniu:

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

Gdzie:

- n - liczba dostępnych przedmiotów,
- p_i - wartość i -tego przedmiotu ($p_i > 0$),
- w_i - waga i -tego przedmiotu ($w_i > 0$),
- x_i - zmienna decyzyjna, $x_i \in \{0, 1\}$ (1 jeśli przedmiot jest wybrany, 0 w przeciwnym razie),
- W - maksymalna dopuszczalna waga plecaka.

2.1 Generowanie Przedmiotów

W ramach projektu zdefiniowano $n = 32$ przedmiotów. Do ich generacji wykorzystano skrypt MATLAB, bazujący na Skrypcie 1 z treści zadania, używając numeru albumu '331129' (niższy numer albumu w przypadku pracy w parze).

- Wagi przedmiotów (w_i) zostały wylosowane z rozkładem równomiernym z przedziału $\langle 0.1, 1 \rangle$ z dokładnością do 0.1.
- Wartości przedmiotów (p_i) zostały wylosowane z rozkładem równomiernym z przedziału $\langle 1, 100 \rangle$ z dokładnością do 1.

Maksymalna pojemność plecaka (W) została ustalona na 30% sumarycznej wagi wszystkich przedmiotów. Suma wag wszystkich przedmiotów wynosi 17.4, zatem $W = 0.30 \times 17.4 = 5.22$.

2.2 Lista Wszystkich Przedmiotów

Poniższa tabela przedstawia listę wszystkich wygenerowanych przedmiotów wraz z ich wagami, wartościami oraz stosunkiem wartości do wagi.

Nr	Waga	Wartość	Stosunek V/W
1	0.2	54	270.00
2	0.2	86	430.00
3	0.5	75	150.00
4	0.6	83	138.33
5	0.2	91	455.00
6	1.0	91	91.00
7	0.7	50	71.43
8	0.6	9	15.00
9	0.2	60	300.00
10	0.8	48	60.00
11	0.2	34	170.00
12	0.7	23	32.86
13	0.3	96	320.00
14	1.0	83	83.00
15	0.2	6	30.00
16	0.4	33	82.50
17	0.2	49	245.00
18	0.8	50	62.50
19	0.6	82	136.67
20	0.7	8	11.43
21	0.5	5	10.00
22	1.0	69	69.00
23	0.9	22	24.44
24	0.7	84	120.00
25	0.7	60	85.71
26	0.8	11	13.75
27	0.4	44	110.00
28	0.8	50	62.50
29	0.1	45	450.00
30	0.5	10	20.00
31	0.5	34	68.00
32	0.4	49	122.50

Tabela 1: Lista wszystkich przedmiotów w problemie plecakowym

3 Algorytm Genetyczny

Do rozwiązania problemu plecakowego wykorzystano algorytm genetyczny dostępny w MATLAB Global Optimization Toolbox.

3.1 Funkcja Celu

Funkcja celu algorytmu genetycznego została zdefiniowana jako minimalizacja negacji sumy wartości wybranych przedmiotów, co jest równoważne maksymalizacji sumy wartości:

$$\text{fitnessFunction} = @(x) - \sum_{i=1}^n x_i \cdot \text{item_values}_i$$

gdzie x jest wektorem binarnym reprezentującym chromosom (wybór przedmiotów).

3.2 Ograniczenia

Jedyne ograniczenie dotyczy sumarycznej wagi wybranych przedmiotów, która nie może przekroczyć maksymalnej pojemności plecaka W :

$$\sum_{i=1}^n x_i \cdot \text{item_weights}_i \leq W_{max}$$

To ograniczenie zostało zaimplementowane jako ograniczenie nierówności liniowej 'Aineq * x' <= bineq' w funkcji 'ga'.

3.3 Parametry Algorytmu Genetycznego

Wybrane parametry algorytmu genetycznego użyte w symulacji to:

- **Rozmiar populacji (PopulationSize):** 100
- **Liczba osobników elitarnych (EliteCount):** 5 (obliczone jako $\lceil 0.05 \times \text{PopulationSize} \rceil$)
- **Współczynnik krzyżowania (CrossoverFraction):** 0.8
- **Funkcja mutacji (MutationFcn):** @mutationuniform, mutationRate
- **Współczynnik mutacji (MutationRate):** 0.04

3.4 Kryteria Zatrzymania Algorytmu

Algorytm genetyczny kończy działanie po spełnieniu jednego z następujących kryteriów:

- **Maksymalna liczba pokoleń (MaxGenerations):** 200
- **Liczba pokoleń bez poprawy (MaxStallGenerations):** 50
- **Tolerancja funkcji celu (FunctionTolerance):** 1e-7

3.5 Konfiguracja Algorytmu w MATLAB

Poniższy fragment kodu przedstawia konfigurację opcji dla algorytmu genetycznego w MATLABie:

```
1 options = optimoptions('ga', ...
2     'PopulationSize', populationSize, ...
3     'EliteCount', eliteCount, ...
4     'CrossoverFraction', crossoverFraction, ...
5     'MutationFcn', {@mutationuniform, mutationRate}, ...
6     'MaxGenerations', maxGenerations, ...
7     'MaxStallGenerations', stallGenerations, ...
8     'FunctionTolerance', functionTolerance, ...
9     'Display', 'off', ...
10    'PlotFcn', {}, ... % Wykresy generowane przez OutputFcn
11    'OutputFcn', @knapsackOutputFcn); % Funkcja zbierająca statystyki
```

Listing 1: Konfiguracja opcji algorytmu genetycznego w MATLAB

Funkcja 'knapsackOutputFcn' została użyta do zbierania statystyk z każdej generacji na potrzeby późniejszej analizy i wizualizacji.

4 Wyniki

Poniżej przedstawiono wyniki uzyskane po uruchomieniu algorytmu genetycznego.

4.1 Rozwiązanie Problemu

Algorytm genetyczny znalazł następujące rozwiązanie:

- **Wektor binarny rozwiązania (x):** Przedmioty wybrane (1) i niewybrane (0) są następujące (indeksy od 1 do 32):
[1 1 1 1 1 1 0 0 1 0 0 0 1 0 0 0 1 0 1 0 0 0 0 1 0 0 0 0 1 0 0 1]
- **Summaryczna waga wybranych przedmiotów:** 5.20 (Maksymalna dozwolona: 5.22)
- **Summaryczna wartość wybranych przedmiotów:** 945

Powyższy wektor binarny został zrekonstruowany na podstawie tabeli wybranych przedmiotów. Summaryczna waga i wartość zostały obliczone na podstawie danych z tej tabeli.

4.2 Wybrane Przedmioty

Tabela poniżej zawiera szczegółowe informacje o przedmiotach wybranych do plecaka.

Nr	Waga	Wartość	Stosunek V/W
1	0.2	54	270.00
2	0.2	86	430.00
3	0.5	75	150.00
4	0.6	83	138.33
5	0.2	91	455.00
6	1.0	91	91.00
9	0.2	60	300.00
13	0.3	96	320.00
17	0.2	49	245.00
19	0.6	82	136.67
24	0.7	84	120.00
29	0.1	45	450.00
32	0.4	49	122.50

Tabela 2: Wybrane przedmioty w rozwiązaniu problemu plecakowego

4.3 Statystyki Wykonania Algorytmu

Informacje o przebiegu algorytmu, takie jak liczba wykonanych pokoleń i liczba obliczeń funkcji celu, są dostępne w strukturze 'output' zwracanej przez funkcję 'ga'.

- **Liczba wykonanych pokoleń:** 70 (wartość z 'output.generations')
- **Liczba obliczeń funkcji celu:** 4234 (wartość z 'output.funccount')
- **Komunikat zakończenia GA:** ga stopped because the average change in the penalty function value is less than options.FunctionTolerance and the constraint violation is less than options.ConstraintTolerance. (treść z 'output.message')

Dokładne wartości tych statystyk mogą nieznacznie różnić się pomiędzy kolejnymi uruchomieniami algorytmu.

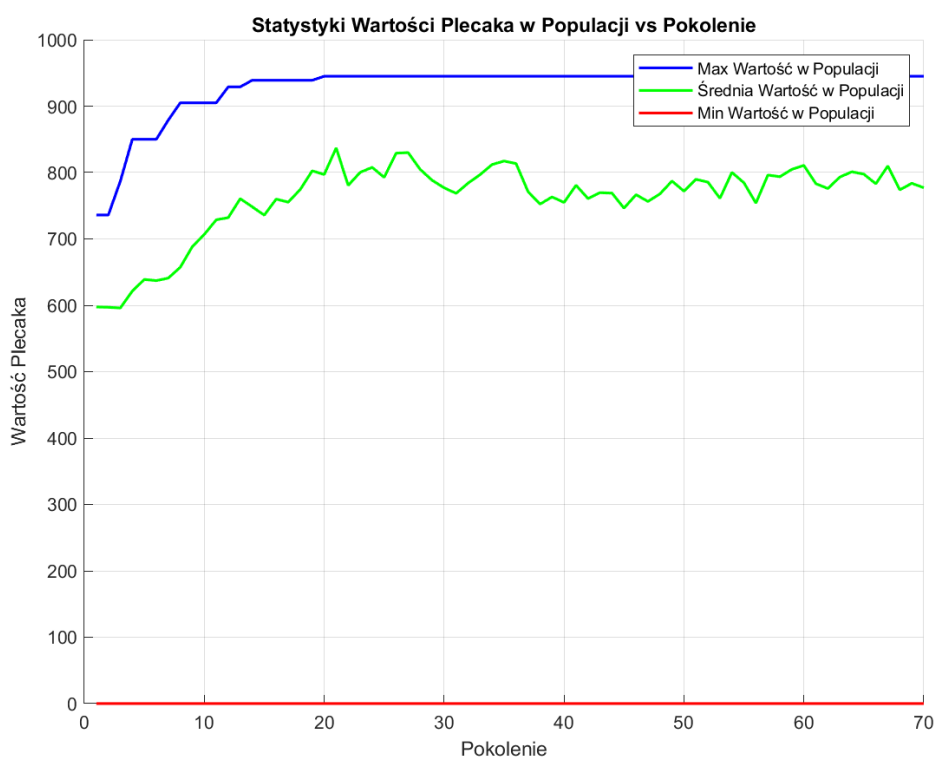
4.4 Liczności Potomków w Każdej Generacji

Na podstawie ustawionych parametrów, liczności potomków w każdej generacji kształtują się następująco:

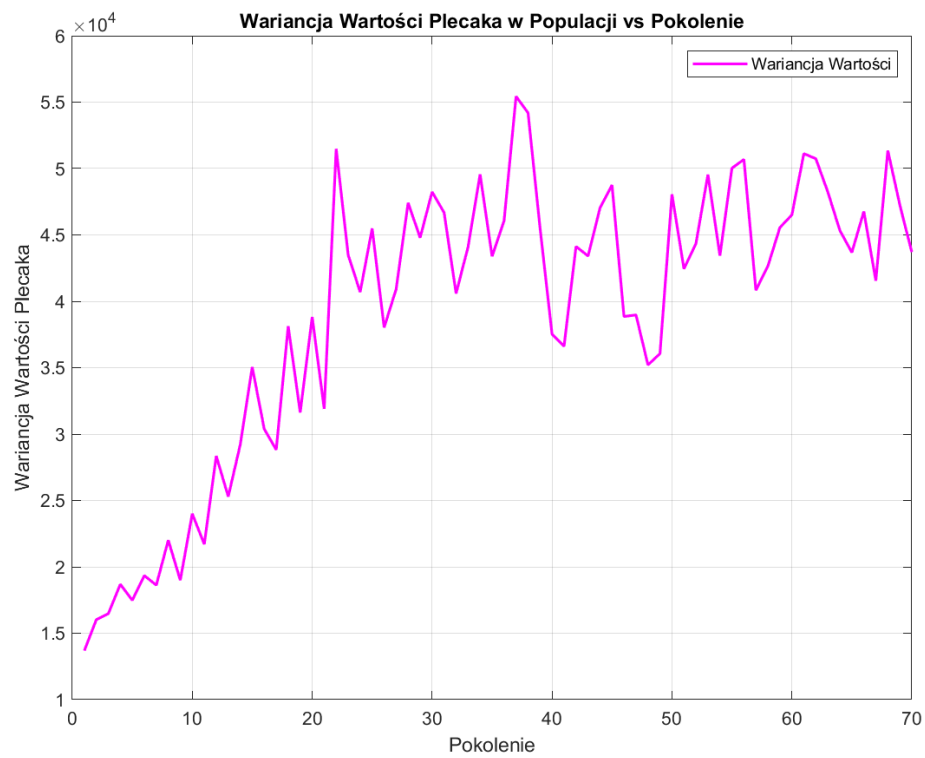
- **Elitarni:** 5
- **Skrzyżowani:** 76 (obliczone jako $\text{round}(\text{CrossoverFraction} \times (\text{PopulationSize} - \text{EliteCount}))$)
- **Zmutowani:** 19 (obliczone jako $(\text{PopulationSize} - \text{EliteCount}) - \text{LiczbaSkrzyżowanych}$)

4.5 Wykresy Przebiegu Algorytmu

Poniższe wykresy ilustrują przebieg algorytmu genetycznego poprzez pokazanie statystyk wartości funkcji celu w kolejnych pokoleniach.



Rysunek 1: Statystyki wartości funkcji celu (Max, Średnia, Min) w populacji w zależności od numeru pokolenia. Wartości na wykresie są rzeczywistymi wartościami plecaka (po odwróceniu minimalizacji).



Rysunek 2: Wariancja wartości funkcji celu w populacji w zależności od numeru pokolenia.

5 Wnioski

Algorytm genetyczny zaimplementowany w środowisku MATLAB okazał się skutecznym narzędziem do rozwiązania problemu plecakowego. Uzyskane rozwiązanie maksymalizuje wartość przedmiotów w plecaku, przestrzegając jednocześnie zadanego ograniczenia wagowego.

- Znalezione rozwiązanie o łącznej wartości 945 i wadze 5.20 mieści się w dopuszczalnym limicie wagowym 5.22.
- Wykresy statystyk funkcji celu (Rysunek 1) pokazują zbieżność algorytmu – średnia i maksymalna wartość funkcji celu w populacji rosną w kolejnych pokoleniach, stabilizując się w późniejszych fazach ewolucji.
- Wykres wariancji (Rysunek 2) ilustruje zmniejszanie się różnorodności populacji w miarę postępu algorytmu, co jest typowe dla procesów ewolucyjnych dążących do optymalnego rozwiązania.

Zastosowane parametry algorytmu genetycznego pozwoliły na znalezienie dobrego jakościowo rozwiązania. Dalsze badania mogłyby obejmować analizę wrażliwości algorytmu na zmiany parametrów oraz porównanie z innymi metodami selekcji czy operatorami genetycznymi w celu potencjalnej dalszej optymalizacji wyników.