



# Problem plecakowy

Mateusz Ferenc (151660), Patryk Górka (158172)

## Wstęp

Problem plecakowy otrzymuje swoją nazwę od problemu maksymalizacji wyboru przedmiotów, gdzie dążymy do osiągnięcia jak największej wartości sumarycznej przy jednoczesnym zachowaniu ograniczenia pojemności plecaka. W ramach tego problemu, mając zbiór elementów z określoną wagą i wartością, naszym celem jest wybranie podzbioru przedmiotów, tak aby suma wartości była maksymalna, jednocześnie nie przekraczając określonej pojemności plecaka. Problem ten należy do klasy problemów **NP-zupełnych**, ponieważ pojemność plecaka nie jest proporcjonalna do rozmiaru danych wejściowych dla problemu.

W tym sprawozdaniu porównywana będzie **efektywność** (czas wykonywania) **algorytmu zachłannego (aproksymacyjnego)** oraz **algorytmu dynamicznego**, w rozwiązywaniu problemu plecakowego. Porówna zostanie również **dokładność** tychże algorytmów.

## Badane algorytmy

### Algorytm wykorzystujący programowanie dynamiczne

Algorytm dynamiczny, stosowany do rozwiązania problemu plecakowego, opiera się na zasadzie podziału problemu na mniejsze podproblemy, rozwiązywaniu ich rekurencyjnie i zapamiętywaniu optymalnych rozwiązań tych podproblemów, aby finalnie skonstruować ostateczny wynik. Dzięki temu gwarantuje znalezienie optymalnego rozwiązania. Implementacja może jednak wymagać przechowywania dużej ilości danych a złożoność czasowa zależy od rozmiaru problemu.

Złożoność obliczeniowa tego algorytmu to  $O(nW)$ , gdzie  $n$  to liczba przedmiotów, a  $W$  to pojemność plecaka. Algorytm ten musi rozważyć wszystkie kombinacje przedmiotów i pojemności plecaka, dlatego jego złożoność jest liniowa względem ilości danych.

### Algorytm zachłanny

Algorytm sortuje przedmioty według wartości jednostkowej (stosunek wartości do wagi) i dodaje tyle przedmiotów, ile tylko może, zaczynając od tych o największej wartości jednostkowej, dopóki nie wypełni plecaka do pełnej pojemności. Algorytm ten dąży do znalezienia rozwiązania bliskiego optymalnemu, wybiera przedmioty o największej wartości jednostkowej, co zwiększa wartość plecaka. Algorytm zachłanny jest prosty i szybki do zaimplementowania, ale nie zawsze zapewnia optymalne rozwiązanie, ponieważ może zignorować kombinacje przedmiotów, które razem dają większą wartość.

Złożoność obliczeniowa tego algorytmu wynosi  $O(n \log n)$ , gdzie  $n$  to liczba przedmiotów. Sortowanie przedmiotów według wartości jednostkowej wymaga czasu  $O(n \log n)$ . Następnie iteruje się przez posortowaną listę przedmiotów, co również zajmuje czas  $O(n)$ . Algorytm zachłanny ma czas działania zbliżony do liniowego, ale zależy od czasu sortowania.

## Metodologia

Algorytmy zostały porównane na dwa sposoby (opisane szczegółowo niżej). Zostały one zaimplementowane w języku python. Biorąc pod uwagę losowość wartości i wag przedmiotów, każdy z testów został powtórzony 20 razy, z czego wyciągnięta została średnia, będąca wynikową wartością.

Czas był mierzony w  $[ms]$ .

Błąd względny obliczany został ze wzoru:  $(D_{dyn} - D_{zach}) / D_{dyn}$ , gdzie  $D$  - maksymalna wartość plecaka używając: **dyn** - algorytmu dynamicznego oraz **zach** - algorytmu zachłannego.

### Stała pojemność plecaka i zwiększająca się liczba elementów

Stała wartość pojemności plecaka wynosiła 1000.

Zarówno czas jak i błąd względny były mierzone dla  $n$  od 100 do 2000 (20 punktów pomiarowych).

Wagi oraz wartości przedmiotów losowane był z przedziału  $< 1; 1000 >$

### Stała liczba elementów i zwiększająca się pojemność plecaka

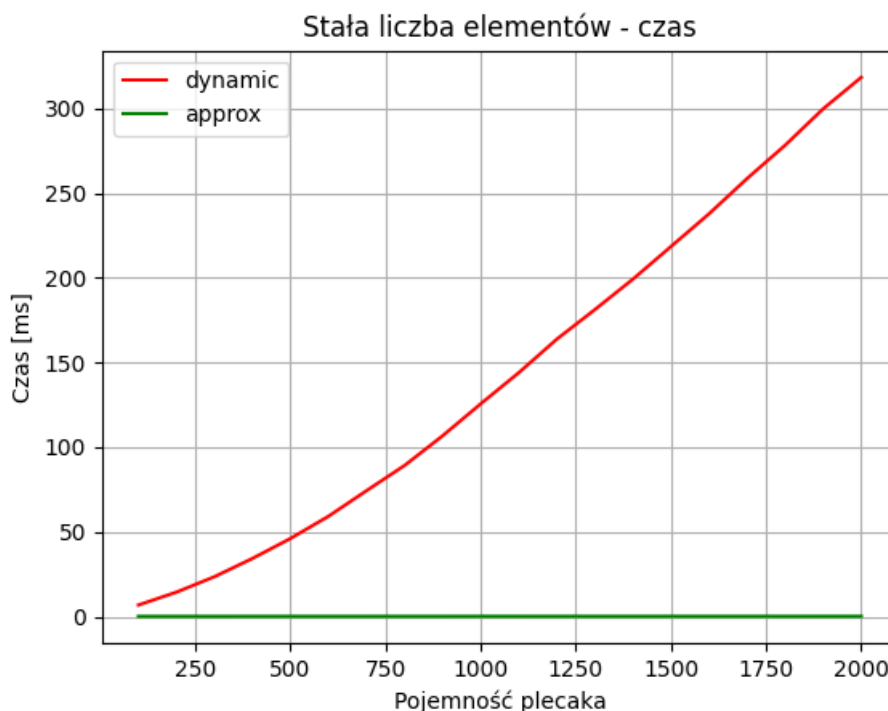
Liczba elementów była stała i wynosiła 750.

Czas oraz błąd względny były mierzone dla  $b$  (maksymalnej pojemności plecaka) od 100 do 2000 (20 punktów pomiarowych).

Wagi oraz wartości przedmiotów losowane był z przedziału  $< 1; 1000 >$

## Wyniki

### Czas działania



Rysunek 1: Czas działania dla stałej liczby elementów

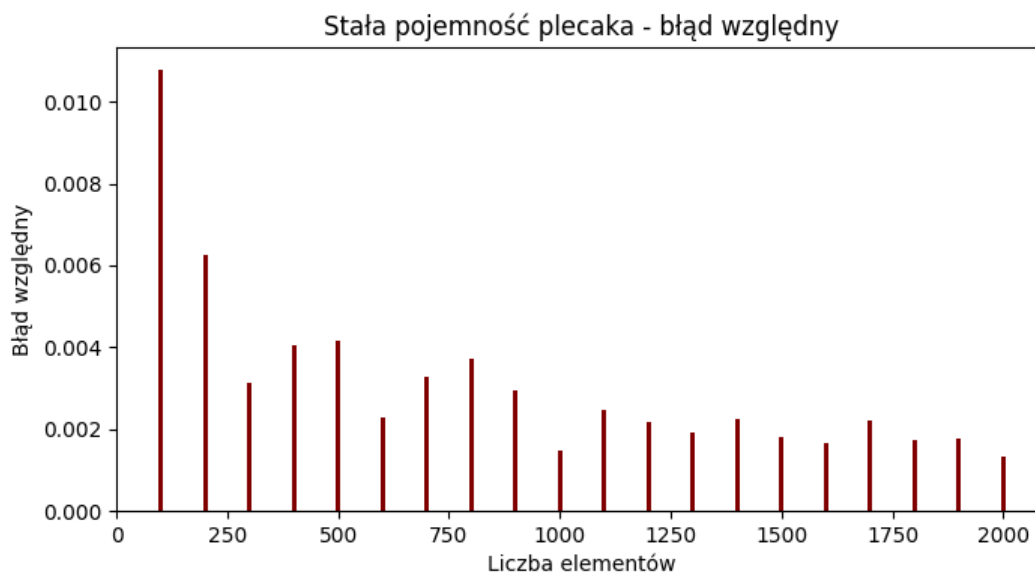


Rysunek 2: Czas działania dla stałej pojemności plecaka

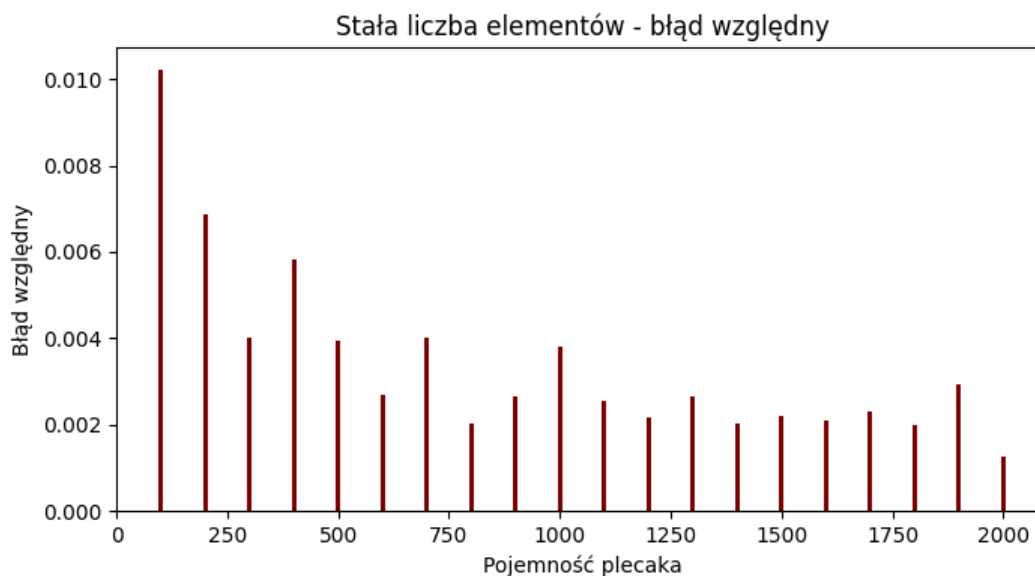
Na pierwszy rzut oka można zauważyć różnicę w złożoności algorytmów - niezależnie od ilości i wielkości wartości przedmiotów, algorytm zachłanny rozwiązuje problem znacznie szybciej niż algorytm dynamiczny.

Czas działania algorytmu wykorzystującego programowanie dynamiczne jest liniowy względem liczby przedmiotów i pojemności plecaka. Oznacza to, że jest on bardziej czasochłonny dla coraz większych instancji problemu. Z kolei algorytm zachłanny działa szybciej, ponieważ jego złożoność jest zbliżona do liniowej.

## Jakość rozwiązań



Rysunek 3: Błąd względny dla stałej pojemności plecaka



Rysunek 4: Błąd względny dla stałej liczby elementów

Jakość rozwiązań uzyskanych przez algorytm dynamiczny jest gwarantowana, ponieważ znajduje on optymalne rozwiązanie. Algorytm zachłanny często daje rozwiązania zbliżone do optymalnych, zwłaszcza gdy różnice między wartościami jednostkowymi przedmiotów nie są znaczące.

Przy większej pojemności plecaka lub liczbie elementów, algorytm zachłanny ma większą szansę na wybór najcenniejszych przedmiotów, co prowadzi do optymalnego rozwiązania. Większa pojemność plecaka daje większą elastyczność w wyborze przedmiotów, natomiast większa liczba dostępnych przedmiotów zwiększa szansę na znalezienie korzystnej kombinacji. Mimo to, algorytm zachłanny może nie dać dokładnie optymalnego wyniku i zawsze istnieje trade-off między szybkością działania a jakością rozwiązania.

## Podsumowanie

Podsumowując, najlepszą metodą rozwiązywania problemu plecakowego jest programowanie dynamiczne. Ta metoda, choć działa wolniej niż algorytm zachłanny, zawsze podaje optymalny wynik, a nie aproksymuje go. Jednak, jeśli możemy sobie pozwolić na przybliżone rozwiązanie, algorytm zachłanny zapewnia je w czasie wielomianowym.