Problem plecakowy (KNAPSACK PROBLEM).

Zagadnienie optymalizacji zwane problemem „plecakowym” swą nazwę wzięło z analogii do sytuacji praktycznej podobnej do problemu pakowania plecaka. Chodzi o to, by zapakować maksymalnie cenny zbiór przedmiotów nie przekraczając ładowności (nośności lub pojemności) bagażu. Można tu wyróżnić kilka sposobów rozumowania. Opis rozpoczynamy od zagadnienia najbardziej ogólnego.

Ogólny problem plecakowy

Dane:

- Przedmioty które oprócz unikatowej nazwy $\mathbf{P_i}$ posiadają dwie cechy:
    1. masę $\mathbf{m_i}$ lub jeśli kto woli objętość (bo pakowanie plecaka możemy optymalizować ze względu na masę (ciężar) lub ze względu na objętość
    2. cenę $\mathbf{c_i}$.

Wyniki:

- ilości $\mathbf{k_i}$ poszczególnych przedmiotów (mogą być zerami)

Formuła problemu:

- Niech:
    - **W** jest wartością wszystkich przedmiotów spakowanych w plecaku.
    - $\mathbf{M_{max}}$ jest maksymalną masą (objętością) plecaka
- Znaleźć wartości $\mathbf{k_i}$, takie aby:
    i. $W = \sum_{i=1}^{n} k_i c_i$ było możliwie największe, oraz
    ii. $M_{max} \geq \sum_{i=1}^{n} k_i m_i$
    iii. $k_i \geq 0$

Rozwiązaniami dopuszczalnymi nazywać będziemy wszystkie zbiory ilości $\mathbf{k_i}$ spełniające warunek ii.

Rozwiązaniem optymalnym będzie ten zbiór ilości $\mathbf{k_i}$, który spełni warunek i.

i - nazwiemy **funkcją optymalizacji**

ii oraz iii - nazwiemy **ograniczeniami** lub **warunkami brzegowymi**.

Zagadnienie w swej prostocie sformułowania wydawać by się mogło łatwym do rozwiązania problemem. Nic bardziej mylnego. Wbrew pozorom problem plecakowy wygenerował wiele dość skomplikowanych algorytmów rozwiązujących go.

Algorytm zachłanny dla ogólnego zagadnienia plecakowego.

Działanie zachłanne przypomina naturalne podejście człowieka do zagadnienia pakowania plecaka. Pakując plecak tak aby pomieścić w nim jak największą wartość człowiek kierowałby się - jednym z lub w sposób mieszany - następującymi kryteriami:

1) Starałby się wybierać rzeczy najcenniejsze w kolejności od najdroższej do najtańszej.

$$c_1 \geq c_2 \geq c_3 \geq \cdots \geq c_n$$

2) Starałby się zabierać rzeczy jak najmniejsze poczynając od najlżejszego przedmiotu, na najcięższym skończywszy.

$$m_1 \leq m_2 \leq m_3 \leq \cdots \leq m_n$$

3) Starałby się wybierać przedmioty w kolejności ustawionej nierosnąco ze względu na iloraz ceny do wagi przedmiotu. Taki iloraz stanowi bowiem jednostkową wartość przedmiotu (wartość przypadającą na jednostkę masy lub objętości).

$$\frac{c_1}{m_1} \geq \frac{c_2}{m_2} \geq \frac{c_3}{m_3} \geq \cdots \geq \frac{c_i}{m_i}$$

Człowiek działając strategicznie i po kolei, podejmuje decyzje optymalne z punktu widzenia danego kroku. Tak więc w pierwszym kroku wg planu nr 1 wziąłby rzecz najdroższą, wg planu nr 2 wziąłby dużo lekkich, zaś wg strategii nr 3 występuje już element wyważenia choć tym razem wziąłby maksymalna ilość przedmiotów o największej wartości jednostkowej.

Właśnie takie działanie nazywamy **działaniem zachłannym**.

Zbadajmy teraz do jakich rezultatów doprowadzi nas postępowanie według każdej z trzech strategii zachłannych podczas pakowania 6 przedmiotów do plecaka o nośności $\mathbf{M_{max}}$=10 jm[1], przy czym każdego przedmiotu mamy dowolną ilość $\mathbf{k_i}$.

Dane:

i	P_i	c_i [zł]	m_i [jm]
1	Koszula flanelowa	75	7
2	Spodnie dżinsowe	150	8
3	Sweter	250	6
4	Czapka baseballowa	35	4
5	Kąpielówki	10	3
6	Obuwie sportowe	100	9

Tabela 1. Dane do analizy rozwiązań problemu plecakowego.

Ad.1

[1] jm – *jednostka miary* – umowna jednostka miary dla potrzeb przykładu.

Zaczynamy od wybrania rzeczy najdroższej. Jest nią sweter, czyli przedmiot o indeksie 3. Sweter waży 6 jm, a więc możemy zapakować ich co najwyżej 1 szt. Zatem $\mathbf{k_3}$=1. Wypełniony swetrami plecak waży 6 jm, musimy zatem jeszcze doładować 4 jm. Najdroższy przedmiot nie przekraczający tej wagi to czapka baseballowa o indeksie 4. Do plecaka zmieści się dokładnie jedna $\mathbf{k_4}$=1 szt. Plecak mamy pełen.

Nasz wynik zatem to:

$$\mathbf{k_1}=0,\ \mathbf{k_2}=0,\ \mathbf{k_3}=1,\ \mathbf{k_4}=1,\ \mathbf{k_5}=0,\ \mathbf{k_6}=0$$

zaś uzyskana wartość plecaka, wynosi:

$$\mathbf{W=k_3c_3+k_4c_4}= 1 * 250 \text{ zł} + 1 * 35 \text{ zł} = 285 \text{ zł}$$

Ad.2

Teraz pakować będziemy „dużo” zaczynając od rzeczy najlżejszej. Najlżejsze w naszym zestawie są kąpielówki o indeksie **i**=5. Masa $\mathbf{m_5}$=3 jm. W plecaku zmieścimy ich aż $\mathbf{k_5}$=3szt.

Plecak nie będzie wypełniony do końca, brakuje 1 jm ale nie ma towaru, który miałby co najwyżej taką masę.

Wartość plecaka w tym przypadku wyniesie:

$$\mathbf{W=k_5c_5}=3 * 10 \text{ zł} = 30 \text{ zł}$$

Jak widać mniej niż w poprzednim przypadku, a więc nie optymalnie. Wypróbujmy teraz sposób trzeci.

Ad.3

Wyliczmy wartości jednostkowe poszczególnych przedmiotów. W tym celu uzupełnimy tabelę 1 o kolumnę c_i/m_i.

i	P_i	c_i [zł]	m_i [jm]	c_i/m_i
1	Koszula flanelowa	75	7	10,71
2	Spodnie dżinsowe	150	8	18,75
3	Sweter	250	6	41,67
4	Czapka baseballowa	35	4	8,75
5	Kąpielówki	10	3	3,33
6	Obuwie sportowe	100	9	11,11

Tabela 2. Wartości jednostkowe.

Plecak zaczniemy zapełniać swetrami, gdyż one mają największą wartość jednostkową. Do plecaka uda nam się zapakować 1 taki sweter. Da to łączną masę 6 jm. Musimy więc zapełnić jeszcze 4 jm. Zmieszczą się jeszcze kąpielówki lub czapka baseballowa. Wybieramy czapkę baseballową (4 jm) gdyż jej wartość jednostkowa jest wyższa od wartości jednostkowej kąpielówek. Sprawdźmy jaką uzyskaliśmy wartość plecaka?

$$\mathbf{W=k_3c_3+k_4c_4}=1 * 250zł + 1 * 35zł = 285\ zł$$

Tym sposobem otrzymaliśmy wartość taką samą jak w metodzie 1, Ale ta metoda wydaje się najbardziej logiczna. Należy uznać, ze metoda optymalizacji poprzez porządkowanie wg. wartości jednostkowej przynosi najlepszy rezultat. Czy jednak jest to wartość optymalna? W naszym przykładzie tak ale można wykazać, że nie zawsze ten sposób rozumowania prowadzi do wyniku optymalnego[2]. Wyniki metod 1 i 2 nie są optymalne zaś działając wg metody 3 często otrzymujemy wynik przybliżony. Mimo to metoda nr 3 wydaje się jedyną spośród rozważanych, godną uwzględnienia w metodach zachłannych rozwiązujących problem plecakowy.

GREEDY-GENERAL-KNAPSACK

Dane:

$P_i(m_i, c_i), i = 1,2, \dots, n;$

gdzie P_i – przedmiot, m_i – masa i-tego przedmiotu, c_i – cena i-tego przedmiotu

Uporządkowane tak, aby: $\frac{c_1}{m_1} \geq \frac{c_2}{m_2} \geq \frac{c_3}{m_3} \geq \cdots \geq \frac{c_i}{m_i}$

M_{max} – nośność (pojemność) plecaka.

Wyniki:

$k_1, k_2, \dots, k_n$ takie, że $k_i \geq 0$ oraz $M_{max} \geq \sum_{i=1}^{n} k_i m_i$

1) Dla kolejnych przedmiotów $P_i,\ i = 1,2,3, \dots, n$ wykonaj krok 2).
2) Określ największą wartość k_i, spełniającą nierówność $k_i m_i \leq M_{max}$
   Przyjmij $M_{max} = M_{max} - k_i m_i$.
3) Znaleziona wartość plecaka wynosi: $W = k_1c_1 + k_2c_2 + \dots + k_nc_n$

   KONIEC

[2] Patrz Maciej M. Sysło *Algorytmy* WSIP Warszawa 2002 s.217-218

Programowanie dynamiczne dla ogólnego problemu plecakowego

Metoda programowania dynamicznego zapewnia znalezienie optymalnego rozwiązania problemu plecakowego. Polega ona na umiejętnym zastosowaniu zasady „dziel i zwyciężaj". Generalnie chodzi o to aby podzielić zagadnienie, na problemy mniejsze – łatwiejsze do rozwiązania. Fajnie byłoby, gdyby pojemność plecaka była mała i mała była także liczba rodzajów przedmiotów do upakowania. No to znajdźmy rozwiązania dla sytuacji jak gdyby trzeba było wypełnić plecak tylko jednym rodzajem przedmiotów. Dodatkowo niech pojemność maksymalna plecaka zmienia się od jednostkowej, do maksymalnej co 1. Wartości wpisywać będziemy w tabeli w której numer wiersza **i** odpowiadać będzie numerowi przedmiotu z tabeli1[3] zaś numer kolumny **j** niech będzie kolejną całkowitą pojemnością plecaka.

Wartość $\mathbf{P_{ij}}$ definiujemy jako wartość optymalnie wypełnionego plecaka o pojemności **j** przedmiotami, których indeksy mieszczą się między 1, a **i**.

Pierwszy wiersz wypełnić najłatwiej albowiem mamy do dyspozycji tylko przedmiot jednego rodzaju, którym napełniamy plecak o kolejnych (1, 2…10) pojemnościach. W kolejne rubryczki wpisujemy wartości plecaka. Pierwszych sześć pozycji ma wartość 0 gdyż koszula waży 7 jm. Dopiero od pojemności plecaka równej 7 możemy ją do niego wpakować.

P_i	c_i	m_i	P_{ij} / i	j: 1	2	3	4	5	6	7	8	9	10
koszula flanelowa	75	7	1	0	0	0	0	0	0	75	75	75	75
spodnie dżinsowe	150	8	2										
sweter	250	6	3										
czapka baseballowa	35	4	4										
kąpielówki	10	3	5										
obuwie sportowe	100	9	6										

Tabela 3a. Tablica P_{ij} wartości upakowań plecaka wygenerowanych przez algorytm programowania dynamicznego – pierwszy wiersz.

W drugim wierszu do dyspozycji mamy już dwa „ciuchy". Niestety w naszym przykładzie ich łączna waga przekracza dopuszczalną ładowność plecaka. Wybieramy zatem ciuch droższy (spodnie dżinsowe 150 zł), który zmieści się w plecaku począwszy od **j**=8. Podobnie w przypadku trzeciego wiersza, gdzie do dyspozycji dochodzi sweter, ale tylko on mieści się w maksymalnym dla każdego **j** plecaku. Wypełnijmy zatem wartościami wiersze 2 i 3.

[3] W algorytmie programowania dynamicznego, kolejność przedmiotów nie ma znaczenia, dlatego tabela nr 1 nie musi być sortowana. Przyp. aut.

P_i	c_i	m_i	P_{ij}	j									
			i	1	2	3	4	5	6	7	8	9	10
koszula flanelowa	75	7	1	0	0	0	0	0	0	75	75	75	75
spodnie dżinsowe	150	8	2	0	0	0	0	0	0	75	150	150	150
sweter	250	6	3	0	0	0	0	0	250	250	250	250	250
czapka baseballowa	35	4	4										
kąpielówki	10	3	5										
obuwie sportowe	100	9	6										

Tabela 3b. Tablica P_{ij} wartości upakowań plecaka wygenerowanych przez algorytm programowania dynamicznego – drugi i trzeci wiersz.

W czwartym wierszu sytuacja się nieco komplikuje. Do plecaka począwszy od pozycji **j**=4 mieści się czapka baseballowa, w pozycji **j**=6 droższy jednak będzie sweter, a w kolumnie **j**=10 zmieści się i czapka i sweter.

P_i	c_i	m_i	P_{ij}	j									
			i	1	2	3	4	5	6	7	8	9	10
koszula flanelowa	75	7	1	0	0	0	0	0	0	75	75	75	75
spodnie dżinsowe	150	8	2	0	0	0	0	0	0	75	150	150	150
sweter	250	6	3	0	0	0	0	0	250	250	250	250	250
czapka baseballowa	35	4	4	0	0	0	35	35	250	250	250	250	285
kąpielówki	10	3	5										
obuwie sportowe	100	9	6										

Tabela 3c. Tablica P_{ij} wartości upakowań plecaka wygenerowanych przez algorytm programowania dynamicznego –czwarty wiersz.

Jeśli poszerzymy asortyment o kąpielówki, to tabela przyjmie kolejną postać:

P_i	c_i	m_i	P_{ij}					j				
			i	1	2	3	4	5	6	7	8	9	10
koszula flanelowa	75	7	1	0	0	0	0	0	0	75	75	75	75
spodnie dżinsowe	150	8	2	0	0	0	0	0	0	75	150	150	150
sweter	250	6	3	0	0	0	0	0	250	250	250	250	250
czapka baseballowa	35	4	4	0	0	0	35	35	250	250	250	250	285
kąpielówki	10	3	5	0	0	10	35	35	250	250	250	260	285
obuwie sportowe	100	9	6										

Tabela 3d. Tablica P_{ij} wartości upakowań plecaka wygenerowanych przez algorytm programowania dynamicznego –piąty wiersz.

I wreszcie wypełniamy ostatni wiersz. Buty – dla każdego **j** wejdą do plecaka tylko one i wcale to nie będzie największa wartość. Zatem szósty wiersz będzie wyglądał tak samo jak piąty.

P_i	c_i	m_i	P_{ij}					j				
			i	1	2	3	4	5	6	7	8	9	10
koszula flanelowa	75	7	1	0	0	0	0	0	0	75	75	75	75
spodnie dżinsowe	150	8	2	0	0	0	0	0	0	75	150	150	150
sweter	250	6	3	0	0	0	0	0	250	250	250	250	250
czapka baseballowa	35	4	4	0	0	0	35	35	250	250	250	250	285
kąpielówki	10	3	5	0	0	10	35	35	250	250	250	260	285
obuwie sportowe	100	9	6	0	0	10	35	35	250	250	250	260	285

Tabela 3. Tablica P_{ij} wartości upakowań plecaka wygenerowanych przez algorytm programowania dynamicznego – kompletnie wypełniona.

Optymalna wartość otrzymana w pozycji $P_{6,10}$ wynosi 285. Jest to ta sama wartość którą otrzymaliśmy stosując algorytm zachłanny, tyle, że tu mamy pewność, że jest to wartość największa.
Podczas wypełniania tabeli rozpoczęliśmy od sytuacji najprostszej. Jeden przedmiot i rosnąca co jeden pojemność plecaka. Jeśli przedmiot nie mieścił się w plecaku, to plecak pozostawał pusty (wartość 0), jeśli natomiast mieścił się, to przypisywaliśmy wartości plecaka krotność wartości przedmiotu (u nas krotność w całym zadaniu wynosiła 1).

W następnym wierszu mogliśmy już wypełniać plecak korzystając z wiersza pierwszego.[4]
Podejmowaliśmy jedną z następujących decyzji:

- wybierz upakowanie rzeczami z wiersza poprzedniego,
- dołóż rzecz o numerze 2 jeśli się zmieści,
- zastąp rzecz nr 1 rzeczą nr 2 jeśli jej wartość jest większa i mieści się tylko jedna spośród nich.

Jeśli $j \geq m_2$ to wybieramy większa spośród $P_{1,j}$ i $P_{2,j-m_2} + c_2$
Podobnie w wierszach następnych.

Tabela Pij przechowuje jedynie wartości optymalnie spakowanego plecaka. Aby móc pokazać zestaw rzeczy składających się na optymalnie spakowany plecak powinniśmy zbudować tabelę Q_{ij} skojarzoną z tabelą P_{ij}, w której przechowywać będziemy indeksy rzeczy pakowanych do plecaka jako ostatnie. To pozwoli na wyznaczenie zestawu rzeczy w plecaku.

P_i	c_i	m_i	Q_{ij}	j									
			i	1	2	3	4	5	6	7	8	9	10
koszula flanelowa	75	7	1	0	0	0	0	0	0	1	1	1	1
spodnie dżinsowe	150	8	2	0	0	0	0	0	0	1	2	2	2
sweter	250	6	3	0	0	0	0	0	3	3	3	3	3
czapka baseballowa	35	4	4	0	0	0	4	4	3	3	3	3	4
kąpielówki	10	3	5	0	0	5	4	4	3	3	3	5	4
obuwie sportowe	100	9	6	0	0	5	4	4	3	3	3	5	4

Tabela 4. Tablica Q_{ij} numerów rzeczy wkładanych do plecaka jako ostatnie.

Na pozycji Q[6,10] znajduje się numer rzeczy włożonej do plecaka na końcu. Rzecz o indeksie i = 4 (czapka baseballowa) waży 4 jm. Zatem cofamy się w wierszu do pozycji Q[6,10-4]=Q[6,6]. Tam zapisano rzecz o indeksie i=3 (sweter), który waży 6 jm. W takim razie powinniśmy przeskoczyć do kolumny 6-6, ale to daje zero (wyczerpała się pojemność plecaka). Zatem w plecaku znajduje się sweter i czapka baseballowa wkładane doń właśnie w takiej kolejności.

DYNAMIC-GENERAL-KNAPSACK

Dane:

$P_i(m_i, c_i), i = 1,2, \dots, n;$

gdzie P_i – przedmiot, m_i – masa i-tego przedmiotu, c_i – cena i-tego przedmiotu
M_{max} – nośność (pojemność) plecaka.

Wyniki:

[4] Zasadę polegającą na podejmowaniu najlepszej decyzji z uwzględnieniem stanu wynikającego z poprzednich decyzji nazywamy **zasadą optymalności Bellmana**.

Tablica wartości $P_{i,j}$ najlepszych upakowań plecaka o pojemności j rzeczami rodzajów od 1 do i; dla i=1,2,….,n oraz j=1,2,… $M_{max.}$

Tablica $Q_{i,j}$ skojarzona z $P_{i,j}$ rzeczy P_i pakowanych do plecaka w ostatnim ruchu.

1) {Ustalenie wartości początkowych tablic P i Q rozszerzonych dla ujednolicenia obliczeń o wiersze i kolumny zerowe.}
   Dla j=1,2,…, M_{max} przypisz $P_{0,j}$:=0, $Q_{0,j}$:=0
   Dla i=1,2,…,n przypisz $P_{i,0}$:=0, $Q_{i,0}$:=0.
2) Dla kolejnych rzeczy i=1,2,…,n wykonaj krok 3.
3) Dla kolejnych pojemności plecaka j=1,2,… M_{max} wykonaj krok 4.
4) Jeśli $j \geq m_i$ {Czyli pojemność plecaka jest wystarczająca, by pomieścić rzecz i} oraz $P_{i-1,j} < P_{i,j-m_i} + c_i$ to przypisz $P_{i-1,j} = P_{i,j-m_i} + c_i$ oraz $Q_{i,j} = i$, a w przeciwnym razie pozostaw wartości z poprzedniego wiersza, czyli przypisz $P_{i,j} = P_{i-1,j}$ oraz $Q_{i,j} = Q_{i-1,j}$.

KONIEC

Decyzyjny problem plecakowy

Problem różni się tym od ogólnego, że każda rzecz pakowana do plecaka może wystąpić tylko jeden raz. Podejmujemy więc decyzję pakować = 1 nie pakować = 0. Tak zdefiniowany problem bardziej odzwierciedla rzeczywistą sytuację pakowania plecaka. Podobnie jak w przypadku możemy mówić o algorytmach zachłannych i dynamicznych które przytaczam tu już bez szerszego omówienia, ze względu na duze podobieństwo do wcześniej omówionych.

GREEDY-DECIDE-KNAPSACK

<u>Dane:</u>

$P_i(m_i, c_i), i = 1,2, \dots, n;$

gdzie P_i – przedmiot, m_i – masa i-tego przedmiotu, c_i – cena i-tego przedmiotu

Uporządkowane tak, aby: $\frac{c_1}{m_1} \geq \frac{c_2}{m_2} \geq \frac{c_3}{m_3} \geq \cdots \geq \frac{c_i}{m_i}$
M_{max} – nośność (pojemność) plecaka.

<u>Wyniki:</u>

$k_1, k_2, \dots, k_n$ takie, że $k_i = 0 \; lub \; k_i = 1$ oraz $M_{max} \geq \sum_{i=1}^{n} k_i m_i$

1. Dla kolejnych rzeczy i = 1,2,…,n wykonaj krok 2.
2. Jeśli $m_i \leq M_{max}$,to przyjmij k_i=1 i przypisz M_{max}= M_{max}-m_i, a w przeciwnym razie przyjmij k_i=0.

3. Utworzony ładunek plecaka ma wartość $W = \sum_{i=1}^{n} k_i c_i$.

KONIEC

DYNAMIC-DECIDE-KNAPSACK

Dane:

$P_i(m_i, c_i), i = 1,2, \dots, n;$

gdzie P_i – przedmiot, m_i – masa i-tego przedmiotu, c_i – cena i-tego przedmiotu
M_{max} – nośność (pojemność) plecaka.

Wyniki:

Tablica wartości $P_{i,j}$ najlepszych upakowań plecaka o pojemności j rzeczami rodzajów od 1 do i; dla i=1,2,….,n oraz j=1,2,… $M_{max.}$

Tablica $Q_{i,j}$ skojarzona z $P_{i,j}$ rzeczy P_i pakowanych do plecaka w ostatnim ruchu.

1) {Ustalenie wartości początkowych tablic P i Q rozszerzonych dla ujednolicenia obliczeń o wiersze i kolumny zerowe.}
   Dla j=1,2,…, M_{max} przypisz $P_{0,j}$:=0, $Q_{0,j}$:=0
   Dla i=1,2,…,n przypisz $P_{i,0}$:=0, $Q_{i,0}$:=0.
2) Dla kolejnych rzeczy i=1,2,…,n wykonaj krok 3.
3) Dla kolejnych pojemności plecaka j=1,2,… M_{max} wykonaj krok 4.
4) Jeśli $j \geq m_i$ {Czyli pojemność plecaka jest wystarczająca, by pomieścić rzecz i} oraz $P_{i-1,j} < P_{i,j-m_i} + c_i$ to przypisz $P_{i-1,j} = P_{i,j-m_i} + c_i$ oraz $Q_{i,j} = 1$, a w przeciwnym razie pozostaw wartości z poprzedniego wiersza, czyli przypisz $P_{i,j} = P_{i-1,j}$ oraz $Q_{i,j} = 0$.

KONIEC

Uwagi końcowe

Złożoność obliczeniowa obydwu algorytmów zachłannych jest proporcjonalna do n $\log_2 n$. Podobnie oba algorytmy dynamiczne mają złożoność obliczeniowa rzędu n M_{max}. Należy zwrócić uwagę, że programowanie dynamiczne będzie sprawdzać się przy niewielkiej ilości elementów i niewielkiej pojemności plecaka. Dla większych liczb algorytmy programowania dynamicznego staja się mało praktyczne.