1 Knapsack Problem - Problem plecakowy

Problem plecakowy jest zagadnieniem optymailzacyjnym. Problem ten swoją nazwę wziął z analogii do rzeczywistego problemu pakowania plecaka. Rozwiązując ten problem zarówno w praktyce jak i teorii trzeba zachować reguły określające ładowność plecaka dotyczące objętości i nośności plecaka. Knapsack Problem zaczął być intensywnie badany po pionierskiej pracy Dantziga[?] w póżnych latach 50 XX wieku. Znalazł on natychmiast zastosowanie w przemyśle oraz w zarządzaniu finansami. Z teoretycznego punktu widzenia, problem plecakowy często występuję jako relaksacja róznorodnych problemów programowania całkowitego[?].

1.1 Zastosowanie

Problem plecakowy stosowany jest nie tylko w sytuacji wynikającej bezpośrednio z nazwy. Znajduje on zastosowanie w wielu dziedzinach życia oraz nauki. Diffi i Helman[?] w 1976 roku oraz Merkle i Helman[?] w 1978 roku zaproponowali problem plecakowy jako podstawę do enkrypcji kluczy prywatnych. Jednakże podejście to w latach późniejszych zostało złamane przez środowisko kryptograficzne i jego miejsce zajęły bardziej odporne algorytmy.

"Knapsack problem" jest stosowany również podczas załadunku kontenerów służacych do przewozu materiałów drogą morską. Ładowność oraz gabaryty ładowanych elementów są ograniczane przez budowę i wytrzymałość kontenera.

Problem ten stosowany jest również w dziedzinie finansów. Jest on podstawowym narzędziem do optymalizacji portfela inwestycyjnego. Poprzez uogólnienie i modyfikacje problemu plecakowego zjawiska ekonomiczne mogą być modelowane z większą dokładnością. Przykładowo możliwe jest zakupienie 0, 1, 2 lub więcej akcji inwestycyjnych, a zakup kolejnych akcji może przynieść obniżenie przychodu.

Wiele problemów związanych z planowaniem może być przyrównana do problemu plecakowego gdzie czas wykonywania operacji na maszynie jest zasobem deficytowym. Jest on szczególnie uwydatniony gdy od aktywności maszyny zależy kapitał przedsiębiorstwa. Poprzez rozwiązanie problemu plecakowego możliwe jest przewidzenie zapotrzebowania na materiały podaczas procesu tak aby warunki zamówinia zostały spełnione[?].

Kolejnym zagadnieniem wynikającym z problemu plecakowego jest problem optymalnego rozkroju, zostanie on przedstawiony w rozdziale ??.

1.2 Różnorodność problemu plecakowego

Wszystkie elementy z rodziny tego problemu wymagają pewnego zestawu elementów które mogą zostać wybrane w taki sposób że zysk zostanie zmaksymalizowany, a pojemość placaka lub plecaków nie zostanie przekroczona.

Wszystkie typy problemu należą do rodziny problemów $NP - trudnych$ co oznacza, że raczej nispotykane jest rozwiązanie problemu z użyciem algorytmów wielomianowych. Możliwe są różne warinaty problemu zależna od rozmieszczenia elementów oraz ilości plecaków[?]:

- *Problem plecakowy 0-1* - każdy element może być wybrany tylko raz. Problem polega na wyborze n elementów dla których suma profitów p_j jest największa, bez konieczności osiągnięcia całkowitej pojemności c. Może być sformułowany jako problem maksymalizacji:

$$\begin{aligned} \text{maksymalizacja} \quad & \sum_{j=1}^{n} p_j x_j, \\ \text{w odniesieniu do} \quad & \sum_{j=1}^{n} w_j x_j \leq c, \\ & x_j \in \{0, 1\}, \qquad j = 1, \ldots, n \end{aligned} \tag{1}$$

  gdzie x_j jest wartością binarną. Jeżeli $x_j = 1$ wtedy j-ty element powinien znaleźć się w plecaku, w innym przypadku $x_j = 0$.

- *Ograniczony problem plecakowy* - każdy element może być wybrany ograniczoną ilość razy. Zmianą w obecnym problemie względem problemu 0-1 jest ograniczona m_j ilość elementów j:

$$\begin{aligned} \text{maksymalizacja} \quad & \sum_{j=1}^{n} p_j x_j, \\ \text{w odniesieniu do} \quad & \sum_{j=1}^{n} w_j x_j \leq c, \\ & x_j \in \{0, 1 \ldots, m_j\}, \quad j = 1, \ldots, n \end{aligned} \tag{2}$$

- *Nieograniczony problem plecakowy* - jest rozszerzeniem problemu ograniczonego o nielimitowaną liczbę dostępnych elementów:

$$\begin{aligned} \text{maksymalizacja} \quad & \sum_{j=1}^{n} p_j x_j, \\ \text{w odniesieniu do} \quad & \sum_{j=1}^{n} w_j x_j \leq c, \\ & x_j \in N_0, \qquad j = 1, \ldots, n \end{aligned} \tag{3}$$

  Każda zmienna x_j w metodzie niograniczonej zostanie ograniczona poprzez pojemność c, gdy waga każdego z elementów jest równa przynajmniej jeden. W ogólnym przypadku transformacja problemu nieograniczonego w ograniczony nie przynosi korzyści

- *Problem plecakowy wielokrotnego wyboru* - elementy powinny być wybierane z klas rozłącznych. Problem ten jest generalizacją problemu 0-1. Możliwy jest wybór dokładnie jednego elementu j z każdej grupy N_i, $i = 1, \ldots, k$:

$$\begin{aligned}
\text{maksymalizacja} \quad & \sum_{i=1}^{k} \sum_{j \in N_i} p_{ij} x_{ij}, \\
\text{w odniesieniu do} \quad & \sum_{i=1}^{k} \sum_{j \in N_i} w_{ij} x_{ij} \leq c, \\
& \sum_{j \in N_i} x_{ij} = 1, \qquad i = 1, \ldots, k, \\
& x_j \in \{0, 1\}, \qquad i = 1, \ldots, k, \quad j \in N_i.
\end{aligned} \tag{4}$$

  Zmienna binarna $x_{ij} = 1$ określa że j-ty element został wybrany z i-tej grupy. Ograniczenie $\sum_{j \in N_i} x_{ij} = 1, \quad i = 1, \ldots, k$ wymusza wybór dokładnie jednego elementu z każdej grupy.

- *Wielokrotny problem plecakowy* - mozliwość wypełnienia wielu plecaków. Jeśli jest możliwość załadowania n elmentów do m pleckaów o róznych pojemnościahc c_i w taki sposób że zysk będzie jak największy:

$$\begin{aligned}
\text{maksymalizacja} \quad & \sum_{i=1}^{k} \sum_{j \in N_i} p_{ij} x_{ij}, \\
\text{w odniesieniu do} \quad & \sum_{j=1}^{n} w_j x_{ij} \leq c_i, \qquad i = 1, \ldots, m \\
& \sum_{j \in N_i} x_{ij} \leq 1, \qquad i = 1, \ldots, k, \\
& x_j \in \{0, 1\}, \qquad i = 1, \ldots, m, \quad j = 1, \ldots, n.
\end{aligned} \tag{5}$$

  Zmienna $x_{ij} = 1$ określa że j-ty element powinien zostać umiesczony w i-tym plecaku, podczas gdy ogranicznie $\sum_{j=1}^{n} w_{ij} x_{ij} \leq c_i$ zapewnia że restrykcja dotycząca pojemności plecaka zostanie zachowana. Ogranicznie $\sum_{j \in N_i} x_{ij} \leq 1$ zapewnia że każdy element zostanie wybrany tylko raz.

- *Bin-packing problem* - bardzo często spotykana wersja problemu plecakowego/ Problem ten polega na umieszczeniu n elementów w jak najmniejszej liczbie opakowań:

$$\begin{aligned}
&\text{maksymalizacja} && \sum_{i=1}^{n} y_i && \\
&\text{w odniesieniu do} && \sum_{j=1}^{n} w_j x_{ij} \leq c y_i, && && i = 1, \ldots, n, \\
& && \sum_{i=1}^{n} x_{ij} = 1, && && j = 1, \ldots, n, \\
& && y_i \in \{0, 1\}, && && i = 1, \ldots, n, \\
& && x_{ij} \in \{0, 1\} && i = 1, \ldots, m, && j = 1, \ldots, n,
\end{aligned} \tag{6}$$

gdzie y_i określa czy i-te opakowanie zostało użyte, a x_{ij} stanowi czy j-ty element powinen zostać umieszczony w i-tym opakowaniu

- *Welokrotnie ograniczony problem plecakowy* - najbardziej ogólny typ który jest problemem programowania całkowitego z dodatnimi współczynnikami:

$$\begin{aligned}
&\text{maksymalizacja} && \sum_{j=1}^{n} p_j x_j, && \\
&\text{w odniesieniu do} && \sum_{j=1}^{n} w_j x_j \leq c_i, && i = 1, \ldots, m, \\
& && x_j \in N_0, && j = 1, \ldots, n.
\end{aligned} \tag{7}$$

1.3 Możliwe rozwiązania

Problem plecakowy należy do grupy problemów $\mathcal{NP}$-Trudnych. Rozwiązanie problemów z tej grupy jest co najmniej tak trudne, jak rozwiązanie każdego problemu z całej klasy $\mathcal{NP}$. Problem $\mathcal{NP}$-Trudny to problem obliczeniowy dla którego znalezienie rozwiązania problemu nie jest możliwe z wielomianową złożonościa obliczeniową. Problemy $\mathcal{NP}$-Trudne obejmują zarówno problemy decyzyjne jak również problemy przeszukiwania czy też problemy optymalizacyjne.

Rozwiązanie problemu plecakowego jest możliwe przy użyciu różnych metod:

- *Metoda podziału i ograniczeń* - Metoda ta często jest stosowana do problemu plecakowego od momentu gdy Kolesar [?] zaprezentował pierwszy algorytm w 1967 roku.

- *Programownaie dynamiczne* - Gdy zostaną dodane warunki brzegowe wtedy algorytm ten staje się "zaawansowaną" formą metody podziału i ograniczeń.

- *Relaksacja przestrzeni stanów* - relaksacja programowani dynamicznego gdzie współczynniki są skalowane przez pewną stałą wartość.

1.3.1 Metoda podziału i ograniczeń

1.3.2 Programowanie dynamiczne

Metoda ta używana jest w przypadku gdy problem można podzielić na małe podproblemy które można rozwiązać rekursywnie. Rozwiązanie optymalne podproblemu jest również optymalnym rozwiązaniem problemu głównego. Przedstawione zostanie rozwiązanie problemu plecakowego rodzaju 0-1.

Jeśli elementy są oznaczone jako $1, \ldots, n$ wtedy podproblem będzie odpowiedzialny za znalezienie optymalnego rozwiązania dla $S_k = \{1, 2, \ldots, k\}$. Niemożliwe jest opisanie rozwiązania końcowego S_n na podstawie podproblemów S_k. Rekursywne sformułowanie podproblemu:

$$B[k, w] = \begin{cases} B[k-1, w] \quad \text{jeśli} \quad w_k > w, \\ max\{B[k-1, w], B[k-1, w-w_k] + b_k\} \quad \text{jeśli} \quad w_k \leq w. \end{cases} \tag{8}$$

Z powyższego równania wynika że najlepszy podzbiór podproblemu S_k z całkowitą wagą w jest najlepszym podzbiorem dla S_{k-1} którego całkowita waga wymosi w lub jest najlepszym podzbiorem dla S_{k-1} którego całkowita waga wynosi $w - w_k$ plus k-ty element. Złożoność programowania dynamicznego to $O(n * W)$. Algorytm jako dane początkowe przyjmuje maksymalną wartość ciężaru W, oraz dwie listy: listę wag $w_1, \ldots, w_n$ orad odpowiadającą jej listę zysku $b_1, \ldots, b_n$.

Algorithm 1 Programowanie dynamiczne - problem plecakowy 0-1

```
for w := 0 TO W do
    B[0,w] := 0
for i := 1 TO n do
    B[i,0] := 0
for i := 1 TO n do
    for w := 0 TO W do
        if w_i <= w then
            if b_i + B[i-1, w-w_i] > B[i-1, w] then
                B[i, w] := b_i + B[i-1, w-w_i]
            else
                B[i, w] := B[i-1, w]
        else
            B[i, w] := B[i-1, w]
```