Przykładowe problemy optymalizacji dyskretnej

❑ Liniowe Modele Optymalizacji Dyskretnej – przykłady dyskretnych problemów decyzyjnych

Zagadnienie rozwózki:

Często mamy do czynienia z sytuacją, gdy pewien jednorodny produkt musi zostać przewieziony od producenta do wielu jego odbiorców. Dla przykładu z cukrowni rozwozi sie wyprodukowany cukier, z mleczarni np. masło i mleko, z browaru piwo itd.

Niekiedy mamy sytuację odwrotną, zwłaszcza w przemyśle spożywczym zakupiony surowiec od wielu jego producentów (np. mleko) należy przewieźć do zakładu (np. zakładów mleczarskich) w którym odbywa się dalsza jego przeróbka.

Tego typu zagadnienia nazywają się ogólnie zagadnieniami rozwózkowo-przywozowymi. Dla uproszczenia w dalszym ciągu będziemy rozważać tylko zagadnienie rozwózki.

❑ Liniowe Modele Optymalizacji Dyskretnej – przykłady dyskretnych problemów decyzyjnych

Zakładamy, że dane są:

- baza będąca miejscem produkcji jednorodnego towaru oraz postoju parku transportowego;

- dana jest liczba pojazdów o jednakowej ładowności;

- znamy popyt każdego odbiorcy;

- oraz macierz odległości (lub kosztu przewozu, czasu przewozu) pomiędzy wszystkimi punktami odbioru;

- popyt każdego odbiorcy jest mniejszy od ładowności pojazdów, a łączne zapotrzebowanie wszystkich punktów odbioru jest mniejsze od ładowności całego parku transportowego;

- towar jest dostarczany do odbiorcy w okresie planowanym (w dniu, tygodniu) przez jeden pojazd.

Należy ustalić taki zbiór marszrut (trasę dostaw towaru) aby:

1. popyt każdego odbiorcy był zrealizowany przez jeden pojazd.

2. ładowność każdego pojazdu nie była przekroczona

3. długość wszystkich marszrut była minimalna

❑ Liniowe Modele Optymalizacji Dyskretnej – przykłady dyskretnych problemów decyzyjnych

Wprowadzamy oznaczenia:

n - liczba odbiorców towaru (punktów odbioru)

P - zbiór indeksów wszystkich punktów odbioru $P = \{1, 2, \ldots, n\}$

$\overline{P}$ - zbiór indeksów wszystkich punktów odbioru i dostawy $\overline{P} = \{0, 1, 2, \ldots, n\}$

m - liczba pojazdów

V - zbiór wszystkich połączeń pomiędzy punktami (możliwych marszrut)

$V = \{\langle i, j \rangle : i, j \in \overline{P} \wedge i \neq j\}$

w - jednakowa ładowność wszystkich pojazdów

b_j - popyt j-tego odbiorcy

c_{ij} - odległość od punktu i do punktu j (długość trasy $\langle i, j \rangle$)

Zgodnie z przyjętymi założeniami dane te muszą spełniać warunki:

$$\sum_{j=1}^{n} b_j \leq m \cdot w \text{ oraz } b_j < w, j \in P$$

Dla każdego pojazdu wyznaczamy jedną marszrutę (łącznie będzie ich m).

Liniowe Modele Optymalizacji Dyskretnej – przykłady dyskretnych problemów decyzyjnych

Przyjmiemy następujące zmienne decyzyjne:

x_{ij} - ilość towaru (dobra) przewożona na trasie $\langle i, j \rangle$

$$y_{ij} = \begin{cases} 1, & \text{gdy pojazd pokonuje trasę } \langle i, j \rangle \\ 0, & \text{w przypadku przeciwnym} \end{cases}$$

Zadanie decyzyjne (PML) będzie miało postać: Znaleźć takie wartości zmiennych x_{ij} oraz y_{ij}, aby:

Funkcja celu: $\sum_{\langle i,j \rangle \in V} c_{ij} \cdot y_{ij} \rightarrow min$

przy warunkach ograniczających:

(1) $\sum_{i \in P} y_{ij} = 1$, dla $(j \in P)$

(2) $\sum_{i \in P} y_{ji} = 1$, dla $(j \in P)$

warunki (1) i (2) oznaczają, że dla każdego odbiorcy wjeżdża i z każdego wyjeżdża jeden pojazd

❑ Liniowe Modele Optymalizacji Dyskretnej – przykłady dyskretnych problemów decyzyjnych

(3) $\sum_{i \in P} y_{i0} = m$

(4) $\sum_{i \in P} y_{0i} = m$

warunki (3) i (4) wymuszają, aby z bazy wyjechało i do niej wróciło dokładnie m - pojazdów

(5) $\sum_{i \in P} x_{ij} - \sum_{i \in P} x_{ji} = b_j, \ (j \in P)$

warunek (5) oznacza, że w każdym punkcie zostawiamy tyle ile wynosi jego popyt

(6) $\sum_{j \in P} x_{0j} = \sum_{j \in P} b_j$

warunek (6) pozwala wywieźć z bazy tyle towaru ile wynosi łączny popyt odbiorców

❑ Liniowe Modele Optymalizacji Dyskretnej – przykłady dyskretnych problemów decyzyjnych

(7) $x_{ij} \le w \cdot y_{ij}, \ \left(\langle i, j \rangle \in V\right)$

warunek (7) zapewnia, że na każdej trasie przewieziemy towaru nie więcej niż wynosi ładowność pojazdu. Jeśli danej trasy pojazd nie pokonuje, to przewóz towaru na tej trasie jest zerowy.

(8) $x_{ij} \ge 0, \ \left(\langle i, j \rangle \in V\right)$

(9) $y_{ij} \in \{0,1\}, \ \left(\langle i, j \rangle \in V\right)$

Zadanie rozwózki jest zadaniem o dużych wymiarach.

liczba zmiennych, to: $L(z) = 2n(n+1)$

liczba warunków: $L(w) = 3n + n(n+1) + 3$

Dla n=30 odbiorców mamy 8450 zmiennych i 483 warunki.

❑ Liniowe Modele Optymalizacji Dyskretnej – przykłady dyskretnych problemów decyzyjnych

Zagadnienie komiwojażera - klasyczny problem optymalizacji dyskretnej

Komiwojażer (dawny sprzedawca objeżdżający domy i oferujący produkty) wyrusza z pewnego miasta (z bazy), ma odwiedzić kilka miejscowości i wrócić do punktu startu. każde z miast może być odwiedzone tylko raz i w dowolnej kolejności.

Dany jest zbiór miast (i=1,2,....,n) oraz nieujemna, kwadratowa macierz odległości (kosztu lub czasu przejazdu) $C = \left[c_{ij} \right] i = 1, \ldots, n; j = 1, \ldots, n$. Należy znaleźć taką drogę zamkniętą, przechodzącą przez wszystkie miejscowości, która jest minimalna.

Droga zamknięta jest zwana dalej marszrutą i składa się z n odcinków, które będziemy nazywać trasami. Ponieważ marszruta nie może zawierać trasy $\langle i, i \rangle$, więc przyjmujemy, że $c_{ii} = \infty$ dla i=1,2,...,n. Łączna liczba marszrut w problemie komiwojażera jest równa $(n-1)!$. Dla n=10 mamy $9! = 362800$ różnych rozwiązań. Przegląd zupełny zbioru rozwiązań w celu znalezienia optymalnego jest efektywny tylko dla małych $n (n \leq 8)$.

❑ Liniowe Modele Optymalizacji Dyskretnej – przykłady dyskretnych problemów decyzyjnych

Oznaczmy przez:

$V = \{\langle i, j\rangle : i \neq j; i, j = 1, 2, \ldots, n\}$ - niech będzie zbiorem wszystkich możliwych tras

Zmienne decyzyjne:

$$x_{ij} = \begin{cases} 1, & \text{gdy marszruta zawiera trasę } \langle i, j\rangle \\ 0, & \text{w przypadku przeciwnym} \end{cases}$$ - zmienna binarna

z_j - zmienna całkowita, która każdemu miastu j-temu przyporządkowuje cechę - kolejność odwiedzenia tego miasta (przy założeni że dla punktu startu - bazy $z_{j_B} = 0$)

Jeżeli n=5 miast oraz $j_B = 1$ (miasto o numerze 1-baza), to przykładowa marszruta może być postaci: $(1, 4, 5, 3, 2, 1)$, a zmienne kolejności odwiedzeń: $z_1 = 0, z_4 = 1, z_5 = 2, z_3 = 3, z_2 = 4$ (oczywiście miasto startu posiadające cechę $z_1 = 0$ nie może mieć drugiej cechy równej 5)

❑ Liniowe Modele Optymalizacji Dyskretnej – przykłady dyskretnych problemów decyzyjnych

Problem komiwojażera sprowadza się do następującego zadania decyzyjnego (PML):

Wyznaczyć takie wartości zmiennych: x_{ij} oraz z_j, aby:

funkcja celu: $\sum_{\langle i,j \rangle \in V} c_{ij} x_{ij} \to \min$

przy warunkach ograniczających:

(1) $\sum_{i=1}^{n} x_{ij} = 1,$ dla $(j = 1, 2, \ldots, n)$

(2) $\sum_{j=1}^{n} x_{ij} = 1,$ dla $(i = 1, 2, \ldots, n)$

warunki (1) i (2) oznaczają, że komiwojażer przez każdy punkt przejeżdża tylko jeden raz

❑ Liniowe Modele Optymalizacji Dyskretnej – przykłady dyskretnych problemów decyzyjnych

(3) $z_i - z_j + nx_{ij} \le n-1$, dla $(i=1,2,\dots,n;\ j=2,3,\dots,n;\ i \ne j)$

niestety (1) i (2) nie gwarantują, że z wybranych n tras tworzymy tylko jedną marszrutę zamkniętą. Warunek (3) wyklucza możliwość powstawania tzw. podcykli w tworzonej marszrucie.

$z_j \ge 0, z_j \in C,\ (j=1,2,\dots,n)$

$x_{ij} \in \{0,1\},\ (\langle i,j \rangle \in V)$

Zadanie komiwojażera jest zadaniem o dużych rozmiarach.

Liczba zmiennych to: $L(z) = n + n(n-1) = n^2$

Liczba warunków to: $L(w) = 2n + n(n-1) - (n-1) = 2n + (n-1)^2$

Dla n=10 mamy 100 zmiennych oraz 101 warunków.

❑ PROGRAMOWANIE SIECIOWE – zadanie komiwojażera (przybliżony algorytm rozwiązania)

Algorytm włączania dla zagadnienia komiwojażera:

Oznaczmy N – zbiór wszystkich miast, zaś przez V – zbiór miast włączonych do marszruty.

Formalnie algorytm ten można opisać w kilku krokach:

1. Podstawić $V := \varnothing$;
2. Wybrać dowolne miasto startowe (bazę, z której wyrusza zaopatrzeniowiec): $i_1 \in N; N := N - \{i_1\}; V := \{i_1\}$;
3. Wybrać zgodnie z pewnym kryterium drugie miasto $i_2 \in N$ tworząc marszrutę: $(i_1, i_2, i_1); N := N - \{i_2\}; V := V + \{i_2\}$;
4. Dla kolejnych miast o numerach k = 3 ,…, n – 1 przeprowadzić operacje:
   - wybrać miasto $i_k \in N$ korzystając z pewnego kryterium i włączyć je do V, czyli: $V := V + \{i_k\}$ - (jest to krok selekcyjny algorytmu),
   - dołączyć miasto i_k do marszruty $(i_1, i_2, ..., i_{k-1}, i_1)$ wstawiając je pomiędzy takie miasta, aby długość powstałej marszruty była największa (jest to krok wstawiania w algorytmie),
5. Dołączyć do marszruty ostatnie n - te miasto stosując to samo postępowanie co dla wcześniejszych miast w kroku 4. Po wykonaniu tych 5 – kroków otrzymuje się marszrutę pełną (V – składa się z n – miast, zaś $N = \varnothing$;

❑ PROGRAMOWANIE SIECIOWE – zadanie komiwojażera (przybliżony algorytm rozwiązania)

Przykład praktyczny:

Rozważmy zadanie komiwojażera z **n = 5** miastami. Macierz – **C** (asymetryczna) określa odległości między tymi miastami:

$$C = \begin{bmatrix} \infty & 2 & 4 & 7 & 3 \\ 2 & \infty & 4 & 5 & 7 \\ 8 & 4 & \infty & 7 & 3 \\ 5 & 9 & 2 & \infty & 7 \\ 4 & 8 & 1 & 4 & \infty \end{bmatrix}$$

- Wybieramy jako miasto początkowe miasto o numerze 5. $V = \{5\}; N = N - \{5\} = \{1,2,3,4\}$.
- W celu wyboru drugiego miasta w marszrucie i każdego kolejnego posłużymy się w kryterium selekcji strategią, która nakazuje wybór miasta położonego **najdalej od aktualnej marszruty niepełnej**.
  Liczne eksperymenty numeryczne potwierdzają dużą efektywność tej strategii.

❑ PROGRAMOWANIE SIECIOWE – zadanie komiwojażera (przybliżony algorytm rozwiązania)

Odległość miasta o **numerze - j** od niepełnej marszruty definiowana jest jako **minimalna odległość** między **j - tym miastem** a **wszystkimi miastami** należącymi do tej marszruty.
Określa to wektor odległości:
$d = (d_1, d_2, ..., d_n)$, gdzie: $d_j = \min\{c_{i,j}\}$; $i \in V; j \in N$;
Dla $j \in V$ (czyli miast należących do marszruty) na **j – tej** pozycji wektora **d** umieszczany jest znak (**minus**).

Dla zadania $d = (4, 8, 1, 4, -)$ - najdalej oddalonym miastem od marszruty jest miasto 2, które dołączamy do marszruty.

Tworzymy marszrutę postaci: $(5, 2, 5)$, której długość wynosi: $F = c_{5,2} + c_{2,5} = 8 + 7 = 15$

$$C = \begin{bmatrix} \infty & 2 & 4 & 7 & 3 \\ 2 & \infty & 4 & 5 & 7 \\ 8 & 4 & \infty & 7 & 3 \\ 5 & 9 & 2 & \infty & 7 \\ 4 & 8 & 1 & 4 & \infty \end{bmatrix}$$

❑ PROGRAMOWANIE SIECIOWE – zadanie komiwojażera (przybliżony algorytm rozwiązania)

- **Dla kolejnych wybieranych miast (krok 4 algorytmu) mamy:**

  a) $V = \{5,2\}; N = N - \{2\} = \{1,3,4\}$

  Wektor $d = \left(\min\{c_{2,1}; c_{5,1}\} = 2, -, \min\{c_{2,3}; c_{5,3}\} = 1, \min\{c_{2,4}; c_{5,4}\} = 4, -\right)$.

  Jako kolejne miasto do marszruty włączamy zatem miasto 4.

  $V = \{5,2,4\}; N = N - \{4\} = \{1,3\}$. Możemy go wstawić pomiędzy miasta (5,2) lub pomiędzy miasta (2,5).

  Jeżeli włączymy między miasta (5,2) utworzymy marszrutę: $(5,4,2,5)$, a tzw. koszt związany z dołączeniem tego miasta wynosi:

  $c_{5,4} + c_{4,2} - c_{5,2} = 4 + 9 - 8 = 5;$

  Jeżeli włączymy między miasta (2,5) utworzymy marszrutę: $(5,2,4,5)$, a tzw. koszt związany z dołączeniem tego miasta wynosi:

  $c_{2,4} + c_{4,5} - c_{2,5} = 5 + 7 - 7 = 5;$

  Włączamy zawsze między takie wierzchołki, dla których koszt włączenia jest najmniejszy. W naszym przypadku koszty są równe, zatem włączamy nowe miasto w miejsce jak najbliższe początkowi marszruty. Marszruta jest więc postaci: $(5,4,2,5)$. Oznacza to, że długość aktualnej marszruty wyniesie $F = F + 5 = 15 + 5 = 20$.

$$C = \begin{bmatrix} \infty & 2 & 4 & 7 & 3 \\ 2 & \infty & 4 & 5 & 7 \\ 8 & 4 & \infty & 7 & 3 \\ 5 & 9 & 2 & \infty & 7 \\ 4 & 8 & 1 & 4 & \infty \end{bmatrix}$$

❑ PROGRAMOWANIE SIECIOWE – zadanie komiwojażera (przybliżony algorytm rozwiązania)

b) $V = \{5,2,4\}; N = \{1,3\}$

Wektor $d = \left(\min\{c_{2,1}; c_{4,1}; c_{5,1}\} = 2, -, \min\{c_{2,3}; c_{4,3}; c_{5,3}\} = 1, -, -\right)$.

Jako kolejne miasto do marszruty włączamy zatem miasto 1.

$V = \{5,2,4,1\}; N = N - \{1\} = \{3\}$. Możemy go wstawić pomiędzy miasta (5,4) (4,2) lub (2,5).

$$C = \begin{bmatrix} \infty & 2 & 4 & 7 & 3 \\ 2 & \infty & 4 & 5 & 7 \\ 8 & 4 & \infty & 7 & 3 \\ 5 & 9 & 2 & \infty & 7 \\ 4 & 8 & 1 & 4 & \infty \end{bmatrix}$$

Jeżeli włączymy między miasta (5,4) utworzymy marszrutę: $(5,1,4,2,5)$, a koszt związany z dołączeniem tego miasta wynosi:

$c_{5,1} + c_{1,4} - c_{5,4} = 4 + 7 - 4 = 7$;

Jeżeli włączymy między miasta (4,2) utworzymy marszrutę: $(5,4,1,2,5)$, a koszt związany z dołączeniem tego miasta wynosi:

$c_{4,1} + c_{1,2} - c_{4,2} = 5 + 2 - 9 = -2$;

Jeżeli włączymy między miasta (2,5) utworzymy marszrutę: $(5,4,2,1,5)$, a koszt związany z dołączeniem tego miasta wynosi:

$c_{2,1} + c_{1,5} - c_{2,5} = 2 + 3 - 7 = -2$;

W naszym przypadku koszty włączenia są najmniejsze w dwóch przypadkach, włączamy miasto 1 pomiędzy miasta (2,5) – bliżej punktu 5 (początek marszruty). Marszruta jest zatem postaci: $(5,4,2,1,5)$. Oznacza to, że długość aktualnej marszruty wyniesie $F = F - 2 = 20 - 2 = 18$.

PROGRAMOWANIE SIECIOWE – zadanie komiwojażera (przybliżony algorytm rozwiązania)

- **Dla ostatniego miasta w marszrucie (krok 5 algorytmu)**

  $V=\{5,2,4,1\}; N=\{3\}$

  Wektor $d=\left(-,-,\min\{c_{1,3};c_{2,3};c_{4,3};c_{5,3}\}=1,-,-\right)$.

  Do marszruty możemy włączyć tylko miasto 3.

  $V=\{5,2,4,1,3\}; N=N-\{3\}=\varnothing$. Możemy go wstawić pomiędzy miasta (5,4) (4,2), (2,1) lub (1,5).

  Dokonujemy porównań kosztów wstawień:

  - marszruta: (5,3,4,2,1,5) - koszt związany z dołączeniem tego miasta wynosi: $c_{5,3}+c_{3,4}-c_{5,4}=1+7-4=4$;

  - marszruta: (5,4,3,2,1,5) - koszt związany z dołączeniem tego miasta wynosi: $c_{4,3}+c_{3,2}-c_{4,2}=2+4-9=-3$;

  - marszruta: (5,4,2,3,1,5) - koszt związany z dołączeniem tego miasta wynosi: $c_{2,3}+c_{3,1}-c_{2,1}=4+8-2=10$;

  - marszruta: (5,4,2,1,3,5) - koszt związany z dołączeniem tego miasta wynosi: $c_{1,3}+c_{3,5}-c_{1,5}=4+3-3=4$;

  W naszym przypadku koszty włączenia są najmniejsze, gdy włączymy miasto 3 pomiędzy miasta (4,2). Pełna marszruta jest zatem postaci: (5,4,3,2,1,5). Oznacza to, że długość tej pełnej marszruty wynosi: $F=F-3=18-3=15$.

$$C=\begin{bmatrix} \infty & 2 & 4 & 7 & 3 \\ 2 & \infty & 4 & 5 & 7 \\ 8 & 4 & \infty & 7 & 3 \\ 5 & 9 & 2 & \infty & 7 \\ 4 & 8 & 1 & 4 & \infty \end{bmatrix}$$

❑ PROGRAMOWANIE SIECIOWE – zadanie komiwojażera (przybliżony algorytm rozwiązania)

Uwagi:

- wybierając jako miasto początkowe inne miasto możemy otrzymać inne rozwiązanie. Zaleca się w tym przypadku wyznaczyć algorytmem przybliżonym rozwiązania dla każdego wierzchołka jako początkowego i wybrać spośród nich najlepsze;
- gdy w danej iteracji w wektorze – d pojawi się więcej niż jeden element maksymalny (można dołączyć więcej niż jedno miasto), to dołączamy miasto o mniejszym numerze;