ALOKAČNÉ ÚLOHY

3. Alokácia výrobných procesov do viacerých miest

3.1 Priradzovací problém (základná verzia)

DEFINÍCIA

Majme n-objektov, ktoré je potrebné umiestniť do n-miest s minimálnymi nákladmi. Ak poznáme náklady c_{ij} (i = 1 ... n, j = 1 ... n) pre lokalizáciu i-teho objektu do j-teho miesta, potom je možné zostaviť jednoduchý bivalentný model.

MODEL:

$x_{i,j} \in \{0,1\}$ -> 1 ak i-ty objekt je umiestnený do j-teho miesta
-> 0 ak i-ty objekt nie je umiestnený do j-teho miesta

$$f(\overline{x}) = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} \overset{!}{=} MIN$$

$$\sum_{i=1}^{n} x_{ij} = 1 \qquad \forall j = 1..n$$

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad \forall i = 1..n$$

3.2 Priradzovací problém (existujú väzby iba medzi novými a existujúcimi objektmi)

DEFINÍCIA

Máme *p*-existujúcich objektov, *n*-nových objektov a *n*-miest. V tomto prípade existujú väzby medzi novými a existujúcimi objektmi. Je známa matica vzdialeností medzi existujúcimi objektmi a novými miestami $\overline{D} = [d_{ij}]_n^P$ a matica prepravných sadzieb $\overline{W} = [w_{ij}]_n^P$ je intenzita väzby medzi starými a novými objektmi. Potom maticu nákladov možno vypočítať nasledovne: $\overline{C} = \overline{W}\,\overline{D} = [c_{ij}]_n^n$.
Bivaletný model je potom rovnaký ako pri základnej verzii priradzovacieho problému.

MODEL:

$x_{i,j} \in \{0,1\}$ -> 1 i-ty objekt je umiestnený do j-teho umiestnenia
-> 0 i-ty objekt nie je umiestnený do j-teho umiestnenia

$$f(\overline{x}) = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} \overset{!}{=} MIN$$

$$\sum_{i=1}^{n} x_{ij} = 1 \qquad \forall j = 1..n$$

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad \forall i = 1..n$$

Príklad:

V tabuľke sú uvedené vzdialenosti v [m] medzi existujúcimi strojmi P, O, R a novými strojmi A, B, C. Z priestorových dôvodov nemožno premiestniť stroj z miesta B do miesta H. Nájdite optimálne rozmiestnenie nových objektov do nových miest E, F, G, H.

Keďže miest je o 1 viac ako strojov, zavedieme fiktívny stroj D s nulovými počtami paliet prepravovanými od neho, resp. k nemu, t.j.:

	Existujuce stroje [ks]	P	O	R
Nove stroje [ks]	A	5	4	2
	B	0	4	3
	C	4	3	2
	D	0	0	0
Mozne miesta [m]	E	1	3	4
	F	4	4	3
	G	5	3	5
	H	6	4	2

$$\overline{C} = \overline{W}\,\overline{D} = \begin{bmatrix} 5 & 4 & 2 \\ 0 & 4 & 3 \\ 4 & 3 & 2 \\ 0 & 0 & 0 \end{bmatrix} . \begin{bmatrix} 1 & 4 & 5 & 6 \\ 3 & 2 & 3 & 4 \\ 4 & 3 & 5 & 2 \end{bmatrix} = \begin{bmatrix} 25 & 34 & 47 & 50 \\ 24 & 17 & 27 & 1000 \\ 21 & 28 & 39 & 40 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Model:

$x_{i,j} \in \{0,1\}$ -> 1 ak i-ty objekt (A,B,C,D) je umiestnený do j-teho miesta (E,F,G,H)

-> 0 ak i-ty objekt nie je umiestnený do j-teho miesta

$$f(\overline{x}) = \sum_{i=1}^{4} \sum_{j=1}^{4} c_{ij} x_{ij} \stackrel{!}{=} MIN$$

$$\sum_{i=1}^{4} x_{ij} = 1 \qquad \forall j = 1..4$$

$$\sum_{j=1}^{4} x_{ij} = 1 \qquad \forall i = 1..4$$

3.3 Kvadratický priradzovací problém

DEFINÍCIA

Máme umiestniť n-nových objektov do n-miest. Medzi novými objektmi existujú vzájomné väzby. Je známa matica vzdialeností $\overline{D} = [d_{ij}]_n^n$, kde d_{ij} je vzdialenosť medzi i-tym a j-tym miestom. A matica prepravných sadzieb $\overline{W} = [w_{ij}]_n^n$, kde w_{ij} je intenzita väzby medzi i-tym a j-tym novým objektom. Je symetrická okolo hlavnej diagonály.

Každé prípustné riešenie možno vyjadriť ako permutáciu $\overline{P} = (p(1), p(2), \ldots, p(n))$, kde $p(i)=k$ znamená, že i-ty objekt pôjde do miesta k.

Náklady pre akúkoľvek permutáciu sú: $f(\overline{P}) = \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} d_{P(i)P(j)}$

METÓDY:

- Metóda CRAFT je heuristická a nezaručí nájdenie najlepšieho riešenia
- Metóda vetvenia a medzí zaručuje nájdenie optimálneho riešenia, ale v nepolynomiálnom čase v závislosti od veľkosti vstupu

CRAFT:

1) Z východzej (náhodnej) permutácie sa vytvorí $\binom{n}{2}$ nových permutácií výmenami všetkých dvojíc objektov v o východzej permutácii.
2) Pre každú permutáciu sa vypočíta kriteriálny funkcia a vyberie sa to najlepšie a stane sa východzou permutáciou pre nasledujúcu iteráciu algoritmu.
3) Celý postup sa opakuje dovtedy, kým sa zlepšuje kriteriálna funkcia z jednej iterácie na druhú.

Príklad:

4 nové stroje môžu byť umiestené do miest A, B, C, D. Vzdialenosti medzi novými miestami sú uvedené v matici $\overline{D}$, denné počty prepravovaných paliet medzi dvojicami nových strojov sú v matici $\overline{W}$. Jednotkové prepravné náklady sú rovnaké.

$$\overline{D} = \begin{bmatrix} 0 & 4 & 5 & 6 \\ 4 & 0 & 3 & 6 \\ 5 & 3 & 0 & 3 \\ 6 & 6 & 3 & 0 \end{bmatrix} \qquad \overline{W} = \begin{bmatrix} 0 & 4 & 1 & 3 \\ 4 & 0 & 2 & 0 \\ 1 & 2 & 0 & 7 \\ 3 & 0 & 7 & 0 \end{bmatrix}$$

Vyjdeme z náhodného rozmiestnenia objektov reprezentovaného napr. permutáciou $\overline{P} = (3,1,4,2)$. Pre takéto rozmiestnenie objektov je hodnota kriteriálnej funkcie:

$$f(\overline{P}) = w_{12}d_{CA} + w_{13}d_{CD} + w_{14}d_{CB} + w_{23}d_{AD} + w_{24}d_{AB} + w_{34}d_{DB} = 4.5 + 1.3 + 3.3 + 2.6 + 0.4 + 7.6 = 86$$

Všetkými možnými výmenami dvojíc objektov vytvoríme nové (susedné permutácie) a pre každú z nich vypočítame hodnotu kriteriálnej funkcie:

Stroj	Stroj	$f(\overline{p})$
1 2 3 4	1 2 3 4	
C A D B	A C D B	86
	D A C B	76
	B A D C	64
	C D A B	66
	C B D A	84
	C A B D	82

Najlepšia hodnota kriteriálnej funkcie v 1. iterácii zodpovedá permutácii (2,1,4,3), s hodnotou 64. Preto táto permutácia sa stane východzou pre nasledujúcu iteráciu:

Stroj	Stroj	$f(\overline{p})$
1 2 3 4	1 2 3 4	
B A D C	A B D C	70
	D A B C	68
	C A D B	86
	B D A C	84
	B C D A	78
	B A C D	68

-> Nie je tu lepšie riešenie ako v 1. iterácii.

Najlepšia hodnota kriteriálnej funkcie je po druhej iterácii 68, čo nie je lepšie, ako hodnota predchádzajúcej permutácie, takže výpočet končí. Výpočet je možné opakovať podľa potreby niekoľkokrát pre ľubovoľné východzie permutácie.

3.4 Zovšeobecnený distribučný problém

DEFINÍCIA

Výrobca dodáva tovar *n*-odberateľom a má k dispozícii konečný počet *m*-miest pre postavenie distribučného centra. Pre každé miesto sú určené fixné náklady f_i spojené so zriadením distribučného centra a prepravné náklady c_{ij} od *i*-teho distribučného centra *j*-temu odberateľovi. Úlohou je vybrať miesta pre zriadenie distribučných centier tak, aby celkové náklady (fixné aj prepravné) boli minimálne.

\+ f_i (pre všetky $i = 1 \dots m$) fixné náklady
c_{ij} (pre všetky $i = 1 \dots$ n, j = 1 ... m)

METÓDY:

A) Celočíselné programovanie
B) Klasický programovací prístup (heuristika)
C) CLP

A) Celočíselné programovanie

y_i (i=1, 2, ..., m) -> 1 ak *i*-te miesto bude vybrané pre zriadenie distribučného centra
-> 0 ak *i*-te miesto nebude vybrané pre zriadenie distribučného centra

$x_{i,j} = \{0, 1\}$ ($i = 1, 2, \dots, m$; $j = 1, 2, \dots, n$)
-> 1 ak *i*-te distribučné centrum bude dodávať *j*-temu odberateľovi
-> 0 ak *i*-te distribučné centrum nebude dodávať *j*-temu odberateľovi

$$f\left(\overline{x}, \overline{y}\right) = \sum_{i=1}^{m} f_i y_i + \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \overset{!}{=} MIN$$

$$\sum_{i=1}^{m} x_{ij} = 1 \qquad \left(j = 1, 2, \dots, n\right)$$

$$x_{ij} \leq y_i \qquad \left(i = 1, 2, \dots, m; \quad j = 1, 2, \dots, n\right)$$

Čo znamená $2^{n+nm}=2^{n(1+m)}$ ohraničení. Pre úlohu reálneho rozmeru (napr. *n*=80, *m*=20) nezvládnuteľný rozmer.

B) Klasický programovací prístup (heuristika)

Pre každý možný výber distribučných centier (tých je 2^m) je priradenie odberateľov triviálne (každého odberateľa preradíme najbližšiemu distribučnému centru). Vývoj programu trval ca. 2 mesiace a pre pomerne malú zmenu zadania je nutné program úplne zmeniť.

C) CLP

Naprogramovanie tej istej úlohy trvalo ca. 2 týždne a bol omnoho flexibilnejší, t.j. napr. zmena zadania si vyžiada jednoduchú zmenu programu. Výborný prototypovací nástroj.

Príklad:

Nesledujú tri verzie riešenia zovšeobecnenej distribučnej úlohy pre tri možné distribučné centrá a piatich odberateľov v deklaratívnom jazyku systému VisualXpress.

1) Distribučné centrá (DC) bez obmedzenia kapacity distribučného centra, každému odberateľovi dodáva práve jedno distribučné centrum.

Program:

```
LET
d=3   !miesta pre distribučné centra
o=5   !odberatelia

TABLES
vzdialenosti(d,o)
fixne_naklady(d)
dodavky(o)

DATA
vzdialenosti(1,1) = 5,3,8,4,2
vzdialenosti(2,1) = 9,6,1,3,5
vzdialenosti(3,1) = 2,4,6,8,3

fixne_naklady(1) = 300, 200, 400
dodavky(1) = 50, 70, 30, 80, 60

VARIABLES
y(d)  !zriadiť, alebo nezriadiť distribučné centrum
x(d,o)      !bude dané DC dodávať danému odberateľovi (áno/nie)

CONSTRAINTS
odberatelia(j=1:o): SUM(i=1:d) x(i,j) = 1 !odberateľ odoberá len od 1 DC
dodavatelia(i=1:d, j=1:o): x(i,j) < y(i)
naklady: SUM(i=1:d) fixne_naklady(i)*y(i) + SUM(i=1:d, j=1:o) dodavky(j)*
    vzdialenosti(i,j)*x(i,j)$

BOUNDS
y(i=1:d) .BV.
x(i=1:d, j=1:o) .BV.
```

2) Distribučné centrá (DC) s obmedzenou kapacitou distribučného centra, každému odberateľovi dodáva práve jedno distribučné centrum.

Program:

```
LET
d=3     !miesta pre distribučné centra
o=5     !odberatelia

TABLES
vzdialenosti(d,o)
fixne_naklady(d)
kapacity(d)
dodavky(o)

DATA
vzdialenosti(1,1) = 5,3,8,4,2
vzdialenosti(2,1) = 9,6,1,3,5
vzdialenosti(3,1) = 2,4,6,8,3

fixne_naklady(1) = 300, 200, 400
kapacity(1) = 150, 130, 170
dodavky(1) = 50, 70, 30, 80, 60

VARIABLES
y(d)  !zriadiť, alebo nezriadiť distribučné centrum
x(d,o)      !bude dané DC dodávať danému odberateľovi (áno/nie)

CONSTRAINTS
odberatelia(j=1:o): SUM(i=1:d) x(i,j) = 1   !odberateľ odoberá len od 1 DC
dodavatelia(i=1:d, j=1:o): x(i,j) < y(i)
kapacita(i=1:d): SUM(j=1:o) x(i,j)* dodavky(j) < kapacity(i)
naklady: SUM(i=1:d) fixne_naklady(i)*y(i) + SUM(i=1:d, j=1:o) dodavky(j)*
    vzdialenosti(i,j)*x(i,j)$

BOUNDS
y(i=1:d) .BV.
x(i=1:d, j=1:o) .BV.
```

3) Distribučné centrá (DC) s obmedzenou kapacitou distribučného centra, jeden odberateľ môže odoberať z viacerých distribučných centier.

Program:

```
LET
d=3     !miesta pre distribučné centra
o=5     !odberatelia

TABLES
vzdialenosti(d,o)
fixne_naklady(d)
kapacity(d)
dodavky(o)

DATA
vzdialenosti(1,1) = 5,3,8,4,2
vzdialenosti(2,1) = 9,6,1,3,5
vzdialenosti(3,1) = 2,4,6,8,3

fixne_naklady(1) = 300, 200, 400
kapacity(1) = 150, 130, 170
dodavky(1) = 50, 70, 30, 80, 60

VARIABLES
y(d)  !zriadiť, alebo nezriadiť distribučné centrum (binárne)
```

```
x(d,o)      !koľko bude dané DC dodávať danému odberateľovi (reálne čísla)

CONSTRAINTS
odberatelia(j=1:o): SUM(i=1:d) x(i,j) = dodavky(j)
kapacita(i=1:d): SUM(j=1:o) x(i,j) < kapacity(i)
naklady: SUM(i=1:d) fixne_naklady(i)*y(i) + SUM(i=1:d, j=1:o)
    vzdialenosti(i,j)*x(i,j)$

BOUNDS
y(i=1:d) .BV.
```