

Přiřazovací problém

Přiřazovací problémy jsou podtřídou logistických úloh, kde lze obecně říci, že m dodavatelů zásobuje m spotřebitelů. Dalším specifikem je, že kapacity dodavatelů (a_i) i požadavky spotřebitelů (b_j) jsou jedničky – úloha je celočíselná.

Příklad:

Jako vhodný příklad tohoto problému převedeného do reálného prostředí lze uvést opravárenskou firmu, kde je pět opravářů. V určitém okamžiku se může objevit současně pět požadavků na opravu. V tomto případě stojíme před problémem, ke kterému klientovi poslat kterého opraváře, aby naše náklady na dopravu byly co nejnižší.

Předpoklady použití tohoto modelu:

- Trasa je kvantifikovatelná
- Jednotky kapacity dodavatelů jsou vzájemně homogenní
- Jednotky požadavků spotřebitelů jsou také vzájemně homogenní
(to znamená, že v rámci dodavatelsko-spotřebitelské vazby je nekonečná mezní míra substituce – lze libovolného j -tého spotřebitele uspokojit dodávkou i -tého dodavatele)
- V případě nepřístupnosti nějakého spoje - osazujeme jej prohibitivní sazbou M

Sazby C_{ij} jsou libovolné a jejich hodnoty mohou vyjadřovat *vzdálenost, čas, ztrátu* nebo *efekt*. U *vzdálenosti, času a ztráty* se snažíme výsledek **minimalizovat** naopak *efekt maximalizovat*.

Celý algoritmus výpočtu přiřazovací úlohy je minimalizační, proto je nutné tabulku upravit, pokud se jedná o hodnoty efektu. Potom převedeme hodnoty tabulky na rozdíly od maximální hodnoty a snažíme se je minimalizovat.

Algoritmus přiřazovací úlohy

- je minimalizační
- je univerzální (platí pro jakoukoliv matici)
- používá se **Maďarská metoda** (König, Egerwarry)
- hledáme právě pět obsazených polí (silně degenerovaná úloha lineárního programování)

Maďarská metoda je založena na této myšlence: místo původní úlohy řešit úlohu s maticí sazeb redukovanou tak, aby všechny sazby zůstaly nezáporné a aby v každém řádku a v každém sloupci byla alespoň jedna sazba nulová. Existuje-li řešení, ve kterém kladným složkám odpovídají nulové sazby, je toto řešení optimální. Neexistuje-li takové řešení, provede se další redukce matice sazeb, atd. Po konečném počtu kroků se dospěje k optimálnímu řešení.

Postup - na matici se díváme jako na celistvý systém

- vytvoříme redukovanou matici, která má identické vlastnosti jako generovaná matice
- **primární redukce** – od každé řady matice sazeb (jak řádek, tak sloupec) odečteme minimální prvek této řady. Je přitom lhostejné, zda začínáme sloupcovými či řádkovými vektory

Generovaná matice		Po řádkové redukci		Po sloupcové redukci																																																																											
<table><tr><td>1</td><td>3</td><td>7</td><td>15</td><td>8</td></tr><tr><td>6</td><td>7</td><td>19</td><td>16</td><td>14</td></tr><tr><td>10</td><td>18</td><td>2</td><td>5</td><td>10</td></tr><tr><td>13</td><td>18</td><td>20</td><td>16</td><td>13</td></tr><tr><td>10</td><td>7</td><td>4</td><td>1</td><td>5</td></tr></table>	1	3	7	15	8	6	7	19	16	14	10	18	2	5	10	13	18	20	16	13	10	7	4	1	5	→	<table><tr><td>0</td><td>2</td><td>6</td><td>14</td><td>7</td></tr><tr><td>0</td><td>1</td><td>13</td><td>10</td><td>8</td></tr><tr><td>8</td><td>16</td><td>0</td><td>3</td><td>8</td></tr><tr><td>0</td><td>5</td><td>7</td><td>3</td><td>0</td></tr><tr><td>9</td><td>6</td><td>3</td><td>0</td><td>4</td></tr></table>	0	2	6	14	7	0	1	13	10	8	8	16	0	3	8	0	5	7	3	0	9	6	3	0	4	→	<table><tr><td>0</td><td>1</td><td>6</td><td>14</td><td>7</td></tr><tr><td>0</td><td>0</td><td>13</td><td>10</td><td>8</td></tr><tr><td>8</td><td>15</td><td>0</td><td>3</td><td>8</td></tr><tr><td>0</td><td>4</td><td>7</td><td>3</td><td>0</td></tr><tr><td>9</td><td>5</td><td>3</td><td>0</td><td>4</td></tr></table>	0	1	6	14	7	0	0	13	10	8	8	15	0	3	8	0	4	7	3	0	9	5	3	0	4
1	3	7	15	8																																																																											
6	7	19	16	14																																																																											
10	18	2	5	10																																																																											
13	18	20	16	13																																																																											
10	7	4	1	5																																																																											
0	2	6	14	7																																																																											
0	1	13	10	8																																																																											
8	16	0	3	8																																																																											
0	5	7	3	0																																																																											
9	6	3	0	4																																																																											
0	1	6	14	7																																																																											
0	0	13	10	8																																																																											
8	15	0	3	8																																																																											
0	4	7	3	0																																																																											
9	5	3	0	4																																																																											

Získali jsme matice, kde v každém řádku nebo sloupci je alespoň jedna nula.

- **nezávislý prvek** – prvek, který vybereme jako samotný (nezávislý pro i -tou řádku a j -tý sloupec)
- **nezávislá nula** (*independent zero*) – vybraná nula, která je sama v řadě
- **krycí čára** (*shadow line*) - vede kolmo na čáru, ve kterém jsme vybrali nulu
- grafický test optima-vylučuje vstup do báze

0	1	6	14	7
0	0	13	10	8
8	16	0	3	8
0	4	7	3	0
9	5	3	0	4

v našem případě v prvním kroku ihned vyšlo optimální řešení (lze dokázat, že nelze nalézt žádné jiné lepší řešení)

→ nezávislá nula

→ krycí čára

Řešení této přiřazovací úlohy nalezneme v původní matici na místech, kde jsou po úpravě nezávislé nuly. V našem případě to je: 24

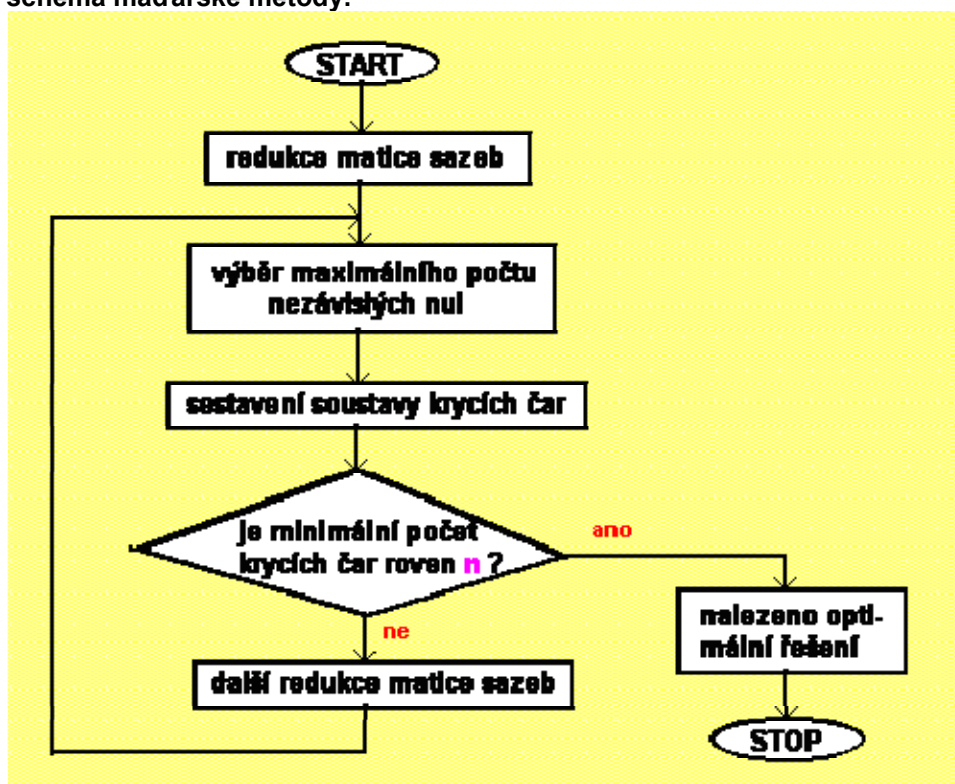
Zadání matice, které bylo náhodně nadiktováno kolegou Lexou, se ukázalo jako řešitelné v jediném kroku. V praxi se však můžeme setkat s případem, že i po splnění veškerých kritérií, se nám nepodaří najít M nezávislých nul. V tomto případě provedeme sekundární redukci.

Sekundární redukce – vybereme minimální $C_{ij} > 0$, které není pokryto a s ním provedeme celou sekundární redukci. V té se postupuje podle stavu v jakém jsou jednotlivé koeficienty C_{ij}

- nepokrytý prvek matice – snížíme o minimální prvek
- prvek pokrytý jednou čarou – neměníme jeho hodnotu
- prvek, kde se křížují dvě čáry – zvýšíme o minimální prvek

Po provedení sekundární redukce se opět pokračuje výběrem nezávislých nul. Na přiloženém obrázku lze vidět zjednodušené schéma průběhu maďarské metody.

Vývojové schéma maďarské metody:



König-Egelwaryho teorém –

minimální počet krycích čar, kterými jsou identifikovány nezávislé nuly tabulky a současně jsou pokryty všechny volné nuly tabulky je roven maximálnímu počtu nezávislých nul, které lze z tabulky vybrat.

Příklad:

(příklad převzat z diplomové práce Petry Klingerové, TU Liberec, 1997)

Šest inženýrů (I₁ až I₆), kteří právě ukončili studium dostalo nabídku šesti pracovišť v různých místech (A až F). Protože každý preferoval pracoviště z různých příčin jinak, sestavili si přehled, v němž každý vyznačil své preference bodováním od jednoho do dvaceti bodů. Výsledky jsou zapsané v tabulce.

	A	B	C	D	E	F
I ₁	4	8	16	20	12	1
I ₂	16	20	8	1	4	12
I ₃	1	12	4	16	20	8
I ₄	4	1	16	12	20	8
I ₅	12	16	1	8	20	4
I ₆	20	16	12	1	4	8

Chceme rozmístění uskutečnit tak, aby součet uspokojených nároků byl maximální. Abychom mohli použít maďarskou metodu, transformujeme danou tabulku na následující matici sazeb:

$$C = \begin{bmatrix} 16 & 12 & 4 & 0 & 8 & 19 \\ 4 & 0 & 12 & 19 & 16 & 8 \\ 19 & 8 & 16 & 4 & 0 & 12 \\ 16 & 19 & 4 & 8 & 0 & 12 \\ 8 & 4 & 19 & 12 & 0 & 16 \\ 0 & 4 & 8 & 19 & 16 & 12 \end{bmatrix}$$

Místo hledání maxima součtu uspokojených nároků budeme hledat minimum součtu diferencí.)

Provedeme redukci matice C (tak, aby po redukci každý řádek a každý sloupec obsahoval alespoň jeden nulový prvek; v našem případě stačí odečíst sloupcová minima ve 3. a 6. sloupci). Dostaneme:

$$C_1 = \begin{bmatrix} 16 & 12 & \cancel{8} & \boxed{0} & 8 & 11 \\ 4 & \boxed{0} & 8 & 19 & 16 & \cancel{3} \\ 19 & 8 & 12 & 4 & \boxed{0} & 4 \\ 16 & 19 & \boxed{0} & 8 & \cancel{3} & 4 \\ 8 & 4 & 15 & 12 & \cancel{3} & 8 \\ \boxed{0} & 4 & 4 & 19 & 16 & 4 \end{bmatrix} \begin{matrix} \\ \\ \times \\ \\ \times \\ \times \end{matrix}$$

Vybrali jsme 5 nezávislých nul a označili 5. řádek, 5. sloupec a 3. řádek. Sestavili jsme soustavu nejmenšího počtu krycích čar - škrtnuli jsme řádek 1., 2., 4., 6. a sloupec 5.

Protože ještě nemůžeme vybrat šest nezávislých nul, vyhledáme nejmenší nepokrytý prvek (v našem případě $\delta = 4$), provedeme další redukci a postup opakujeme.

Dostaneme:

$$C_2 = \begin{bmatrix} 16 & 12 & \cancel{8} & \boxed{0} & 12 & 11 \\ 4 & \boxed{0} & 8 & 19 & 20 & \cancel{3} \\ 15 & 4 & 8 & \cancel{3} & \cancel{3} & \boxed{0} \\ 16 & 19 & \boxed{0} & 8 & 4 & 4 \\ 4 & \cancel{3} & 11 & 8 & \boxed{0} & 4 \\ \boxed{0} & 4 & 4 & 19 & 20 & 4 \end{bmatrix}$$

V matici C_2 již lze vybrat 6 nezávislých nul (dvěma způsoby). Vybereme je v pořadí:

1. varianta		2. varianta	
řádek	sloupec	řádek	sloupec
6	1	6	1
4	3	4	3
1	4	1	4
2	2	2	6
5	5	5	2
3	6	3	5

První možnost je v tabulce vyznačena orámováním nezávislých nul. Úloha má tedy 2 optimální řešení.

1. varianta	A	B	C	D	E	F	2. varianta	A	B	C	D	E	F
I ₁	0	0	0	1	0	0	I ₁	0	0	0	1	0	0
I ₂	0	1	0	0	0	0	I ₂	0	0	0	0	0	1
I ₃	0	0	0	0	0	1	I ₃	0	0	0	0	1	0
I ₄	0	0	1	0	0	0	I ₄	0	0	1	0	0	0
I ₅	0	0	0	0	1	0	I ₅	0	1	0	0	0	0
I ₆	1	0	0	0	0	0	I ₆	1	0	0	0	0	0

Řešení odpovídá těmto rozmístěním:

inženýr	1. varianta		2. varianta	
	pracoviště	preference	pracoviště	preference
I ₁	D	20	D	20
I ₂	B	20	F	12
I ₃	F	8	E	20
I ₄	C	16	C	16
I ₅	E	20	B	16
I ₆	A	20	A	20
součet uspokojených nároků		104		104