

Flöde i nätverk

Det ligger högar av "viktiga saker" på några platser, och de ska transporteras till andra platser. Kostnaderna är linjära.

Variabeldefinition: x_{ij} = flöde i båge (i, j) .

Bågdata för båge (i, j) :

- c_{ij} : flödeskostnad per enhet.
- u_{ij} : övre gräns för flödet.
- l_{ij} : undre gräns för flödet.

Bivillkor: $l_{ij} \leq x_{ij} \leq u_{ij}$

Noddata för nod i :

- b_i : källstyrka/sänkstyrka. (måste vara givet)

Nodjämviktsvillkor: $\sum_{j:(j,i) \in B} x_{ji} - \sum_{j:(i,j) \in B} x_{ij} = b_i$ för alla $i \in N$ (in - ut)

Krav på indata: $\sum_i b_i = 0$.

Flöde i nätverk

Sats

Varje anslutningsmatris är fullständigt unimodulär.

Slutsats

Flödesproblem kan betraktas som LP-problem. Flödet blir automatiskt heltal.

Obs: Inga andra bivillkor får finnas.

Minkostnadsflödesproblemet

Skicka efterfrågade mängder så billigt som möjligt.

$$\begin{aligned} \min \quad & \sum_{(i,j) \in B} c_{ij} x_{ij} \\ \text{då} \quad & \sum_{j:(j,i) \in B} x_{ji} - \sum_{j:(i,j) \in B} x_{ij} = b_i \text{ för alla } i \in N \\ & l_{ij} \leq x_{ij} \leq u_{ij} \text{ för alla } (i,j) \in B \end{aligned}$$

Specialfall av minkostnadsflödesproblemet

Billigaste väg-problemet: Minkostnadsflödesproblem med en källa och en sänka, båda av styrka ett.

$$b_i = \begin{cases} -1 & \text{då } i = s \\ 1 & \text{då } i = t \\ 0 & \text{f ö} \end{cases}$$

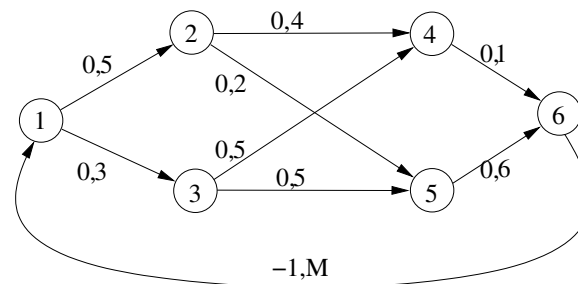
$l_{ij} = 0$ och u_{ij} stor (men i praktiken 1) för alla bågar.

Riktade brevbärarproblemet: Cirkulerande minkostnadsflödesproblem med undre gräns ett för alla bågar.

Specialfall av minkostnadsflödesproblemet

Maxflödesproblemet: Inför återbåge (t, s) . Sätt $c_{ts} = -1$ och $c_{ij} = 0$ för alla $(i, j) \in B \setminus (ts)$. Sök cirkulerande flöde (inga källor eller sänkor).

$$\begin{aligned} \max \quad & f \\ \text{då} \quad & \sum_{j:(j,i) \in B} x_{ji} - \sum_{j:(i,j) \in B} x_{ij} = \begin{cases} -f & \text{då } i = s \\ f & \text{då } i = t \\ 0 & \text{f ö} \end{cases} \text{ för alla } i \in N \\ & 0 \leq x_{ij} \leq u_{ij} \text{ för alla } (i,j) \in B \\ & f \text{ fri} \end{aligned}$$



Specialfall av minkostnadsflödesproblemet

Transportproblemet: Minkostnadsflödesproblem i tudelad graf.

$$\begin{aligned} \min \quad & \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \\ \text{då} \quad & \sum_{j=1}^n x_{ij} = s_i \quad i = 1, \dots, m \\ & \sum_{i=1}^m x_{ij} = d_j \quad j = 1, \dots, n \\ & x_{ij} \geq 0 \quad \text{för alla } i, j \end{aligned}$$

Kan ses som ett matrisproblem, där i står för rader och j för kolumner. Bivillkoren specificerar radsummor och kolumnsummor.

(Eftersom detta är ett minkostnadsflödesproblem blir lösningen heltal.)

Lösningmetod för minkostnadsflödesproblemet

$$\begin{aligned} \min \quad & \sum_{(i,j) \in B} c_{ij} x_{ij} \\ \text{då} \quad & \sum_{j:(j,i) \in B} x_{ji} - \sum_{j:(i,j) \in B} x_{ij} = b_i \quad \text{för alla } i \in N \\ & l_{ij} \leq x_{ij} \leq u_{ij} \quad \text{för alla } (i,j) \in B \end{aligned}$$

Simplexmetoden

Baslösning:

Icke-basvariabler: $x_{ij} = l_{ij}$ eller $x_{ij} = u_{ij}$.
(Övre och undre gränser behandlas implicit.)

Hur många basvariabler?

Ett av nodjämviktsvillkoren är redundant. En $n \times n$ -matris vore linjärt beroende. Kan bara ha bas av dimension $n - 1$. Alltså $n - 1$ basvariabler.

En cykel motsvarar linjärt beroende. Alltså ingen cykel i basen.

Slutsats: En bas motsvarar ett **uppspännande träd**.

Specialfall av minkostnadsflödesproblemet

Tillordningsproblemet: Transportproblem med all tillgång och efterfrågan lika med ett.

Varje person i skall göra en uppgift och varje uppgift j ska göras en gång.

Variabeldefinition: $x_{ij} = 1$ om person i gör uppgift j .

$$\begin{aligned} \min \quad & \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \\ \text{då} \quad & \sum_{j=1}^n x_{ij} = 1 \quad i = 1, \dots, n \\ & \sum_{i=1}^n x_{ij} = 1 \quad j = 1, \dots, n \\ & x_{ij} \geq 0 \quad \text{för alla } i, j \end{aligned}$$

Lösningmetod: **Ungerska metoden**. Baseras på LP-dualen.

(Eftersom detta är ett minkostnadsflödesproblem blir lösningen heltal.)

Simplexmetoden för minkostnadsflödesproblemet

En iteration:

Basvariablerna ger ett *uppspännande träd*.

Inkommande variabel bildar en unik *cykel*.

Man vill skicka runt så mycket som möjligt i cykeln.

Utgående variabel begränsar flödesändringen i cykeln. Den hamnar på övre eller undre gräns.

Utgående variabel bryter upp cykeln.

Simplexmetoden för minskostnadsflödesproblemet

Hur välja inkommande variabel?

y : dualvariabler till nodjämviktsvillkoren. y_i kallas nodpris för nod i .

Reducerade kostnader: $\hat{c}_{ij} = c_{ij} + y_i - y_j$.

(Nod i till j : Jämför skillnaden i nodpris, $y_j - y_i$, med direktvägen, c_{ij} .)

Minimering

$\hat{c}_{ij} < 0 \Rightarrow$ Båge (i, j) billig. \Rightarrow Skicka så mycket som möjligt. \Rightarrow Öka x_{ij} .

$x_{ij} = u_{ij}$ är optimalt.

$\hat{c}_{ij} > 0 \Rightarrow$ Båge (i, j) dyr. \Rightarrow Skicka så lite som möjligt. \Rightarrow Minska x_{ij} .

$x_{ij} = l_{ij}$ är optimalt.

Optimalitetsvillkor

$\hat{c}_{ij} < 0 \Rightarrow x_{ij} = u_{ij}$, (billig ickebas)

$\hat{c}_{ij} > 0 \Rightarrow x_{ij} = l_{ij}$ (dyr ickebas)

samt $l_{ij} < x_{ij} < u_{ij} \Rightarrow \hat{c}_{ij} = 0$ (bas). (Används för att räkna ut y .)

Simplexmetoden för minskostnadsflödesproblemet

0. Finn tillåten startbas (träd).

1. Sätt $y_1 = 0$. Beräkna resterande y via basträdet. ($c_{ij} = y_j - y_i$ för alla basbågar.)

2. Beräkna reducerade kostnader $\hat{c}_{ij} = c_{ij} + y_i - y_j$ för alla icke-basbågar.

3. Kontrollera optimalitet. Om optimum: Stopp.

4. Välj mest lovande variabel som inkommande.

5. Finn cykeln som bildas.

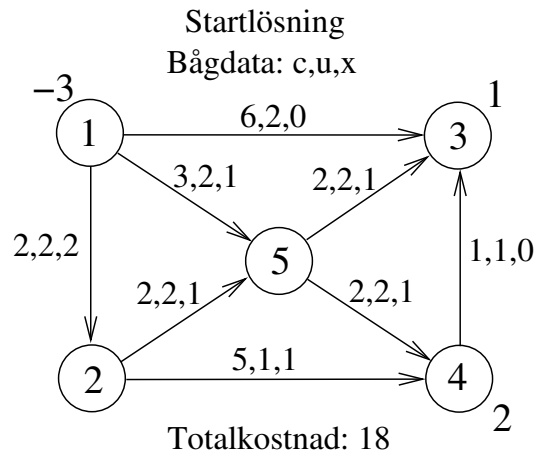
6. Ändra flödet i cykeln maximalt i önskad riktning.

7. Välj den variabel som begränsar ändringen som utgående variabel.

8. Gå till 1.

Simplexmetoden för minskostnadsflöde: Exempel

Indata:



Simplexmetoden för minskostnadsflödesproblemet

Känslighetsanalys

Ändring av kostnad för icke-basbåge eller införande av ny båge:

Beräkna ny reducerad kostnad, $\hat{c}_{ij} = c_{ij} + y_i - y_j$ och kontrollera optimalitet.

Tillordningsproblemet

Varje person i skall göra en uppgift och varje uppgift j ska göras en gång.

Variabeldefinition: $x_{ij} = 1$ om person i gör uppgift j .

$$\begin{aligned} \min \quad & z = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \\ \text{då} \quad & \sum_{j=1}^n x_{ij} = 1 \quad i = 1, \dots, n \\ & \sum_{i=1}^n x_{ij} = 1 \quad j = 1, \dots, n \\ & x_{ij} \geq 0 \quad \text{för alla } i, j \end{aligned}$$

Metod: Formulera och lös LP-dualen.

Arbeta med dualvariabler och reducerade kostnader.

Ungerska metoden för tillordningsproblemet

Arbeta med reducerade kostnader:

$$\hat{c}_{ij} = c_{ij} - \alpha_i - \beta_j \text{ för alla } (i, j).$$

Duala bivillkoren (primal optimalitet): $\hat{c}_{ij} \geq 0$ för alla (i, j)

Komplementaritetsvillkoren: $\hat{c}_{ij} x_{ij} = 0$ för alla (i, j)

Optimalitetsvillkor: $\hat{c}_{ij} > 0 \Rightarrow x_{ij} = 0$, $x_{ij} > 0 \Rightarrow \hat{c}_{ij} = 0$

Tillåtna positioner: där $\hat{c}_{ij} = 0$.

Mål: Placera exakt en etta i varje rad och i varje kolumn på de tillåtna positionerna.

Tillordningsproblemet: Lös LP-dualen

$$\begin{aligned} \min \quad & z = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \\ \text{då} \quad & \sum_{j=1}^n x_{ij} = 1 \quad i = 1, \dots, n \\ & \sum_{i=1}^n x_{ij} = 1 \quad j = 1, \dots, n \\ & x_{ij} \geq 0 \quad \text{för alla } i, j \end{aligned}$$

$$\text{LP-dual: } \max v = \sum_{i=1}^n \alpha_i + \sum_{j=1}^n \beta_j \quad \text{då } \alpha_i + \beta_j \leq c_{ij} \text{ för alla } (i, j).$$

Komplementaritetsvillkor: $x_{ij}(\alpha_i + \beta_j - c_{ij}) = 0$ för alla (i, j) .

Metod: Lös dualen: Öka α och β , utan att överskrida duala bivillkoren.

Försök sedan finna en tillåten primallösning som uppfyller komplementaritetsvillkoren. Ändra α och β om det inte går.

Ungerska metoden för tillordningsproblemet

Lös dualen.

- Sätt $\alpha_i = 0$ för alla i och $\beta_j = 0$ för alla j (vilket ger $\hat{c}_{ij} = c_{ij}$).
- Öka enstaka α_i och β_j så mycket som möjligt (utan att \hat{c}_{ij} blir negativt).
- Kan en etta placeras i varje rad och kolumn på de tillåtna positionerna? Om ja, stopp.
- Om inte, ändra dualvariablerna i par:
Öka α_i och minska β_j så att inget \hat{c}_{ij} blir negativt.

Gör detta så att nya tillåtna positioner, $\hat{c}_{ij} = 0$, bildas, samtidigt som $\hat{c}_{ij} \geq 0$ för alla i, j .

Upprepa tills tillåten lösning fås.

Ungerska metoden för tillordningsproblemet

Matris av reducerade kostnader, \hat{C} .

Stryk alla tillåtna positioner med minsta möjliga antal streck.

Sats

Högsta antalet ettor som kan placeras ut är lika med minsta antalet streck som behövs.

Öka α_i för ostrukna rader, och minska β_j för strukna kolumner.

Effekt: Ostrukna element minskas, enkelt strukna element oförändrade, dubbelt strukna element ökas.

\hat{c}_{ij} minskas inte för någon tillåten position.

Inget \hat{c}_{ij} blir negativt.

Minst en ny tillåten position bildas.

Ungerska metoden: Exempel

$$\hat{c}_{ij} = c_{ij} - \alpha_i - \beta_j$$

					α
β	0	0	0	1	3
	1	1	0	0	6
	0	0	-1	0	2
	0	-2	5	1	5
	0	1	-1	1	

Till slut återstår bara en nolla.

Ungerska metoden för tillordningsproblemet

0. Starta med de ursprungliga kostnaderna ($\alpha = 0, \beta = 0$).

1. Dra bort det minsta elementet i varje rad från alla elementen i raden (öka α).

2. Dra bort minsta elementet i varje kolumn från alla elementen i kolumnen (öka β).

3. Stryk alla nollor med minsta möjliga antal streck.

Om antalet streck är lika med n , finn tillåten lösning, och sluta.

4. Dra bort minsta ostrukna element från alla ostrukna element och addera till alla dubbelt strukna element. (Öka α_i för ostrukna rader, och minska β_j för strukna kolumner.) Gå till 3.

Ungerska metoden: Exempel

$$\text{Optimallösning: } x = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

Person 1 gör uppgift 2, person 2 gör uppgift 3, person 3 gör uppgift 4, person 4 gör uppgift 1.

Total kostnad: 17

Duallösning:

$$\alpha = (3, 6, 2, 5)$$

$$\beta = (0, 1, -1, 1)$$

Kolla gärna starka dualsatsen: $v = \sum_{i=1}^n \alpha_i + \sum_{j=1}^n \beta_j$

Summan av dualvariablerna ska vara lika med totala kostnaden.