Flöde i nätverk

Flöde i nätverk

Det ligger högar av “viktiga saker” på några platser, och de ska transporteras till andra platser.

Flöde i nätverk

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

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).

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.

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

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.

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)

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)

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.

Flöde i nätverk

Sats

Varje anslutningsmatris är fullständigt unimodulär.

Slutsats

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

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.

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

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.

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.

$$\min \quad \sum_{(i,j)\in B} c_{ij}x_{ij}$$

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

$$l_{ij} \le x_{ij} \le u_{ij} \quad \text{för alla } (i,j) \in B$$

Specialfall av minkostnadsflödesproblemet

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

Specialfall av minkostnadsflödesproblemet

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

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

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.

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).

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).

$$\max \quad f$$

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

$$0 \leq x_{ij} \leq u_{ij} \quad \text{för alla } (i, j) \in B$$

$$f \text{ fri}$$

Specialfall av minkostnadsflödesproblemet

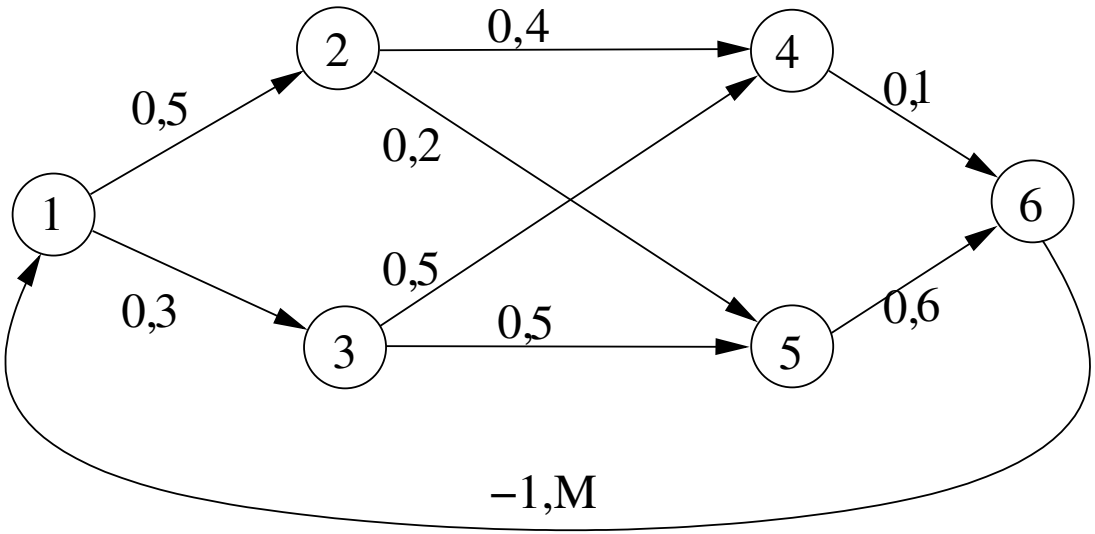
Maxflödesproblemet: Inför återbåge (t, s). Sätt $c_{ts} = -1$ och $c_{ij} = 0$ för alla $(i, j) \in B \setminus (ts)$. Sök cirkulerande flöde (inga källor eller sänkor).

$$\max \quad f$$

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

$$0 \le x_{ij} \le u_{ij} \quad \text{för alla } (i, j) \in B$$

f fri



Specialfall av minkostnadsflödesproblemet

Transportproblemet: Minkostnadsflödesproblem i tudelad graf.

Specialfall av minkostnadsflödesproblemet

Transportproblemet: Minkostnadsflödesproblem i tudelad graf.

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

Specialfall av minkostnadsflödesproblemet

Transportproblemet: Minkostnadsflödesproblem i tudelad graf.

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

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

Specialfall av minkostnadsflödesproblemet

Transportproblemet: Minkostnadsflödesproblem i tudelad graf.

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

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.)

Specialfall av minkostnadsflödesproblemet

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

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.

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.

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.

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

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.

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

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

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.

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

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

(Eftersom detta är ett minkostnadsflödesproblem blir lösningen heltal.)

Utvidgning av minkostnadsflödesproblemet

Flervaruflöde: Flödet består av olika sorters varor med separat tillgång/efterfrågan men med gemensamma kapaciteter.

Utvidgning av minkostnadsflödesproblemet

Flervaruflöde: Flödet består av olika sorters varor med separat tillgång/efterfrågan men med gemensamma kapaciteter.

Även för olika start- och slutnodsbundna signaler, som i telekomnät.

Utvidgning av minkostnadsflödesproblemet

Flervaruflöde: Flödet består av olika sorters varor med separat tillgång/efterfrågan men med gemensamma kapaciteter.

Även för olika start- och slutnodsbundna signaler, som i telekomnät.

Variabeldefinition: x_{ij}^k = flöde av sort k i båge (i, j).

Utvidgning av minkostnadsflödesproblemet

Flervaruflöde: Flödet består av olika sorters varor med separat tillgång/efterfrågan men med gemensamma kapaciteter.

Även för olika start- och slutnodsbundna signaler, som i telekomnät.

Variabeldefinition: x_{ij}^k = flöde av sort k i båge (i,j).

$$\min \quad \sum_{(i,j)\in B} \sum_{k\in C} c_{ij}^k x_{ij}^k$$

$$\text{då} \quad \sum_{j:(j,i)\in B} x_{ji}^k - \sum_{j:(i,j)\in B} x_{ij}^k = b_i^k \quad \text{för alla } i \in N, k \in C$$

$$\sum_{k\in C} x_{ij}^k \leq u_{ij} \quad \text{för alla } (i,j) \in B$$

$$l_{ij}^k \leq x_{ij}^k \leq u_{ij}^k \quad \text{för alla } (i,j) \in B, k \in C$$

Utvidgning av minkostnadsflödesproblemet

Flervaruflöde: Flödet består av olika sorters varor med separat tillgång/efterfrågan men med gemensamma kapaciteter.

Även för olika start- och slutnodsbundna signaler, som i telekomnät.

Variabeldefinition: x_{ij}^k = flöde av sort k i båge (i,j).

$$\min \quad \sum_{(i,j)\in B} \sum_{k\in C} c_{ij}^k x_{ij}^k$$

$$\text{då} \quad \sum_{j:(j,i)\in B} x_{ji}^k - \sum_{j:(i,j)\in B} x_{ij}^k = b_i^k \quad \text{för alla } i \in N, k \in C$$

$$\sum_{k\in C} x_{ij}^k \le u_{ij} \qquad \text{för alla } (i,j) \in B$$

$$l_{ij}^k \le x_{ij}^k \le u_{ij}^k \qquad \text{för alla } (i,j) \in B, k \in C$$

Bivillkorsmatrisen är *ej* fullständigt unimodulär.

Lösningsmetod för minkostnadsflödesproblemet

$$\min \quad \sum_{(i,j)\in B} c_{ij}x_{ij}$$

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

$$l_{ij} \leq x_{ij} \leq u_{ij} \quad \text{för alla } (i,j) \in B$$

Lösningsmetod för minkostnadsflödesproblemet

$$\min \quad \sum_{(i,j)\in B} c_{ij}x_{ij}$$

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

$$l_{ij} \leq x_{ij} \leq u_{ij} \quad \text{för alla } (i,j) \in B$$

Simplexmetoden

Lösningsmetod för minkostnadsflödesproblemet

$$\min \quad \sum_{(i,j)\in B} c_{ij}x_{ij}$$

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

$$l_{ij} \leq x_{ij} \leq u_{ij} \quad \text{för alla } (i,j) \in B$$

Simplexmetoden

Baslösning:

Lösningsmetod för minkostnadsflödesproblemet

$$\min \quad \sum_{(i,j)\in B} c_{ij}x_{ij}$$

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

$$l_{ij} \leq x_{ij} \leq u_{ij} \quad \text{för alla } (i,j) \in B$$

Simplexmetoden

Baslösning:

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

Lösningsmetod för minkostnadsflödesproblemet

$$\min \quad \sum_{(i,j)\in B} c_{ij}x_{ij}$$

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

$$l_{ij} \leq x_{ij} \leq u_{ij} \quad \text{för alla } (i,j) \in B$$

Simplexmetoden

Baslösning:

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

Hur många basvariabler?

Lösningsmetod för minkostnadsflödesproblemet

$$\min \sum_{(i,j)\in B} c_{ij}x_{ij}$$

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

$$l_{ij} \le x_{ij} \le u_{ij} \text{ för alla } (i,j) \in B$$

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.

Lösningsmetod för minkostnadsflödesproblemet

$$\min \sum_{(i,j)\in B} c_{ij}x_{ij}$$

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

$$l_{ij} \le x_{ij} \le u_{ij} \text{ för alla } (i,j) \in B$$

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.

Lösningsmetod för minkostnadsflödesproblemet

$$\min \sum_{(i,j)\in B} c_{ij}x_{ij}$$

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

$$l_{ij} \le x_{ij} \le u_{ij} \text{ för alla } (i,j) \in B$$

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**.

Simplexmetoden för minkostnadsflödesproblemet

En iteration:

Simplexmetoden för minkostnadsflödesproblemet

En iteration:

Basvariablerna ger ett *uppspännande träd*.

Simplexmetoden för minkostnadsflödesproblemet

En iteration:

Basvariablerna ger ett *uppspännande träd*.

Inkommande variabel bildar en unik *cykel*.

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.

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.

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 minkostnadsflödesproblemet

Hur välja inkommande variabel?

Simplexmetoden för minkostnadsflödesproblemet

Hur välja inkommande variabel?

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

Simplexmetoden för minkostnadsflödesproblemet

Hur välja inkommande variabel?

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

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

Simplexmetoden för minkostnadsflödesproblemet

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}.)

Simplexmetoden för minkostnadsflödesproblemet

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

Simplexmetoden för minkostnadsflödesproblemet

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.

Simplexmetoden för minkostnadsflödesproblemet

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.

Simplexmetoden för minkostnadsflödesproblemet

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

$\hat{c}_{ij} < 0 \Rightarrow x_{ij} = u_{ij}$,

Simplexmetoden för minkostnadsflödesproblemet

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)

Simplexmetoden för minkostnadsflödesproblemet

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

Simplexmetoden för minkostnadsflödesproblemet

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)

Simplexmetoden för minkostnadsflödesproblemet

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 $l_{ij} < x_{ij} < u_{ij} \Rightarrow \hat{c}_{ij} = 0$

Simplexmetoden för minkostnadsflödesproblemet

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).

Simplexmetoden för minkostnadsflödesproblemet

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 minkostnadsflödesproblemet

0. Finn tillåten startbas (träd).

Simplexmetoden för minkostnadsflödesproblemet

0. Finn tillåten startbas (träd).

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

Simplexmetoden för minkostnadsflödesproblemet

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.

Simplexmetoden för minkostnadsflödesproblemet

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.

Simplexmetoden för minkostnadsflödesproblemet

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.

Simplexmetoden för minkostnadsflödesproblemet

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.

Simplexmetoden för minkostnadsflödesproblemet

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.

Simplexmetoden för minkostnadsflödesproblemet

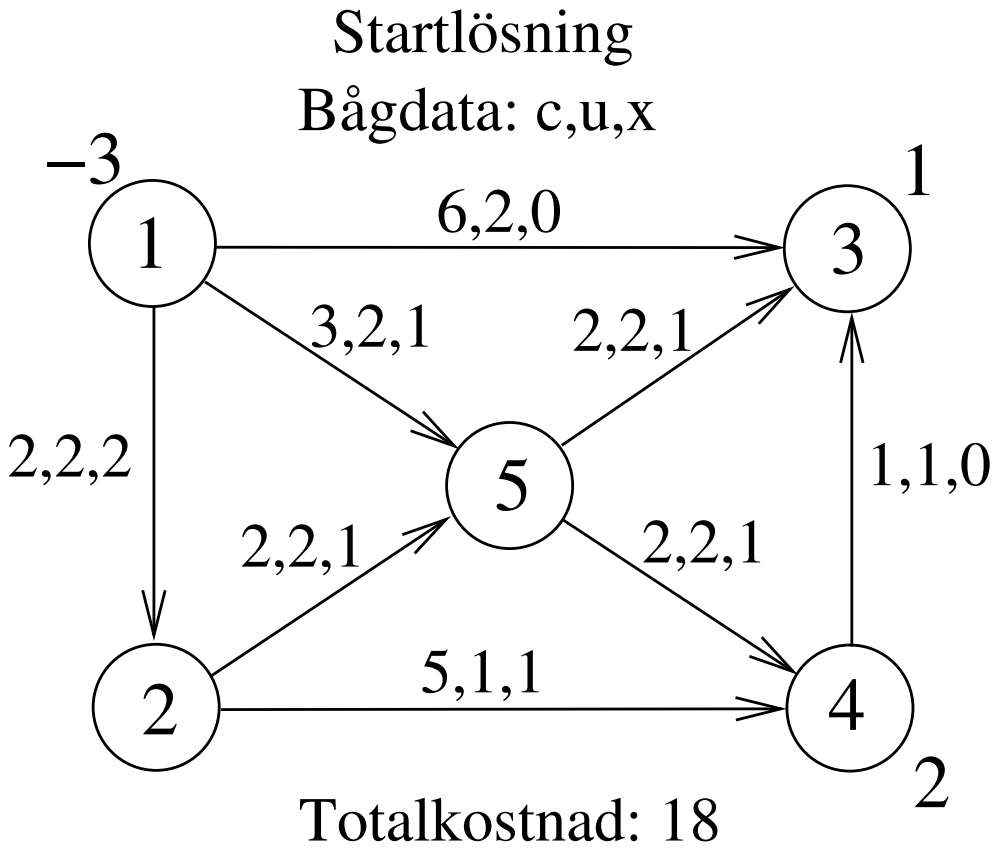
0. Finn tillåten startbas (träd).
1. Sätt $y_1 = 0$. Beräkna resterande y via basträdet. ($c_{ij} = y_j - y_i$ för alla basbågar.)
2. Beräkna reducerade kostnader $\hat{c}_{ij} = c_{ij} + y_i - y_j$ för alla icke-basbågar.
3. Kontrollera optimalitet. Om optimum: Stopp.
4. Välj mest lovande variabel som 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.

Simplexmetoden för minkostnadsflödesproblemet

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.

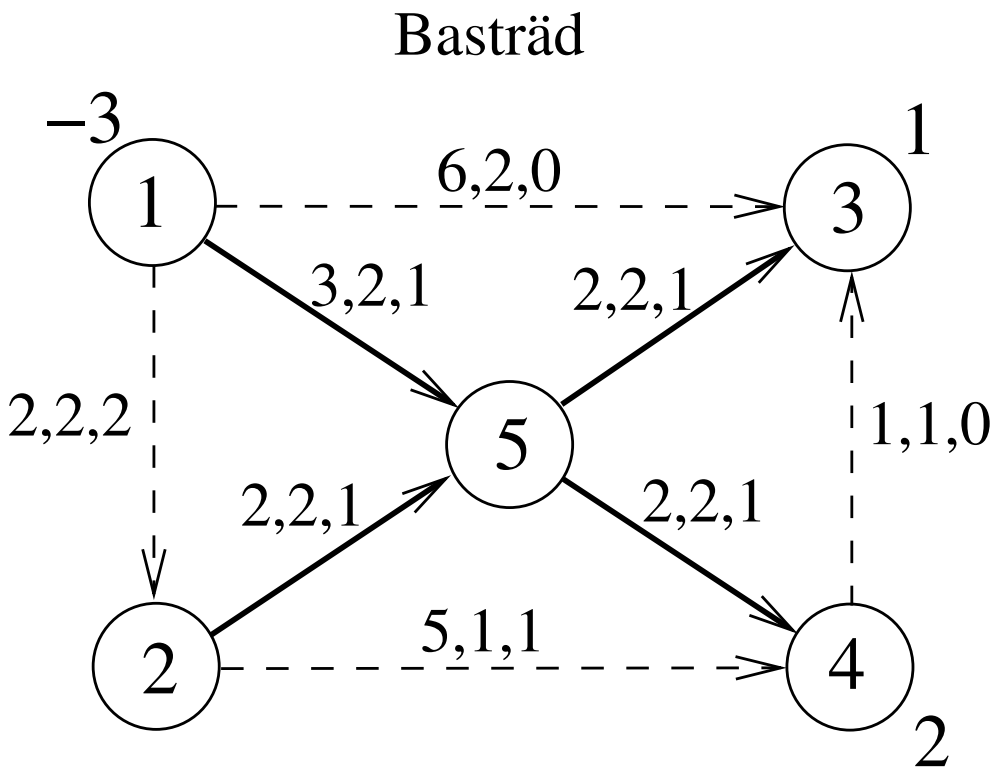
Simplexmetoden för minkostnadsflöde: Exempel

Indata:



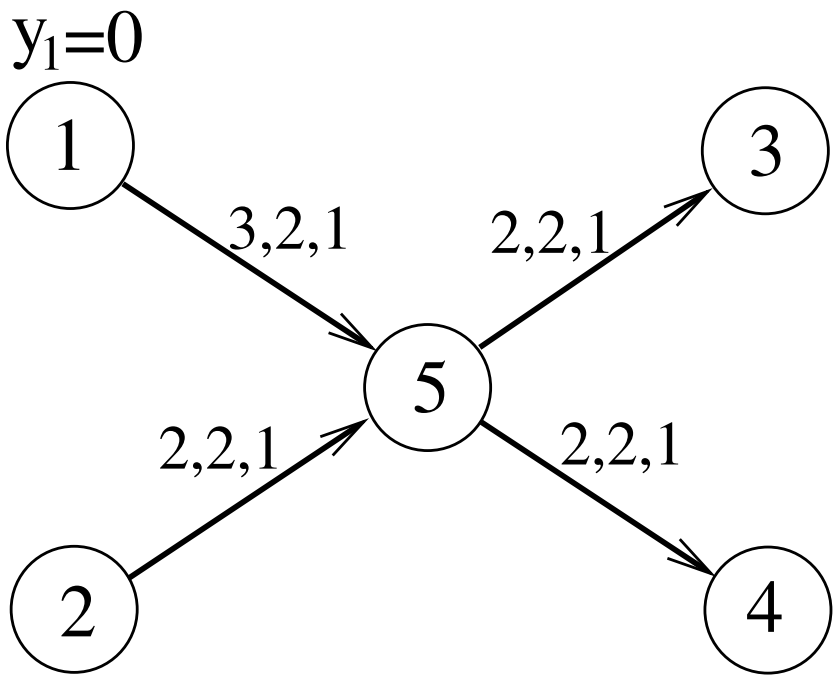
Simplexmetoden för minkostnadsflöde: Exempel

x inte på gräns: basbåge. Måste ge uppspännande träd.



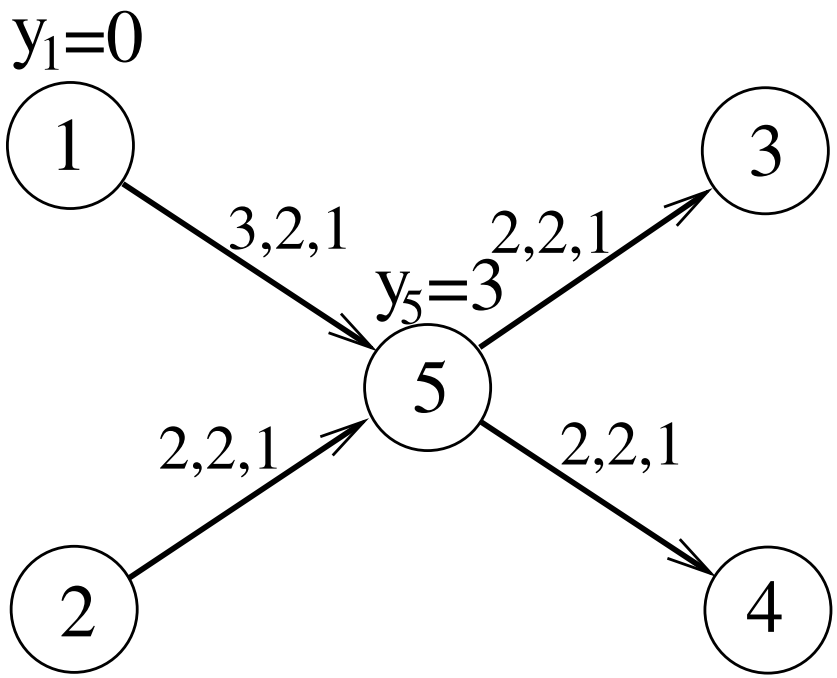
Simplexmetoden för minkostnadsflöde: Exempel

Räkna ut y via basbågarna.



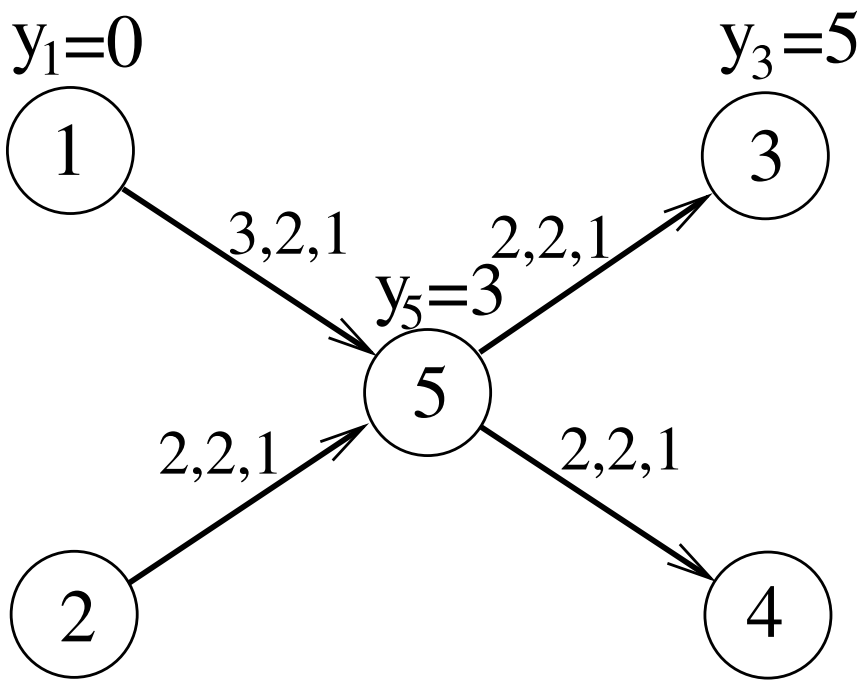
Simplexmetoden för minkostnadsflöde: Exempel

Räkna ut y via basbågarna.



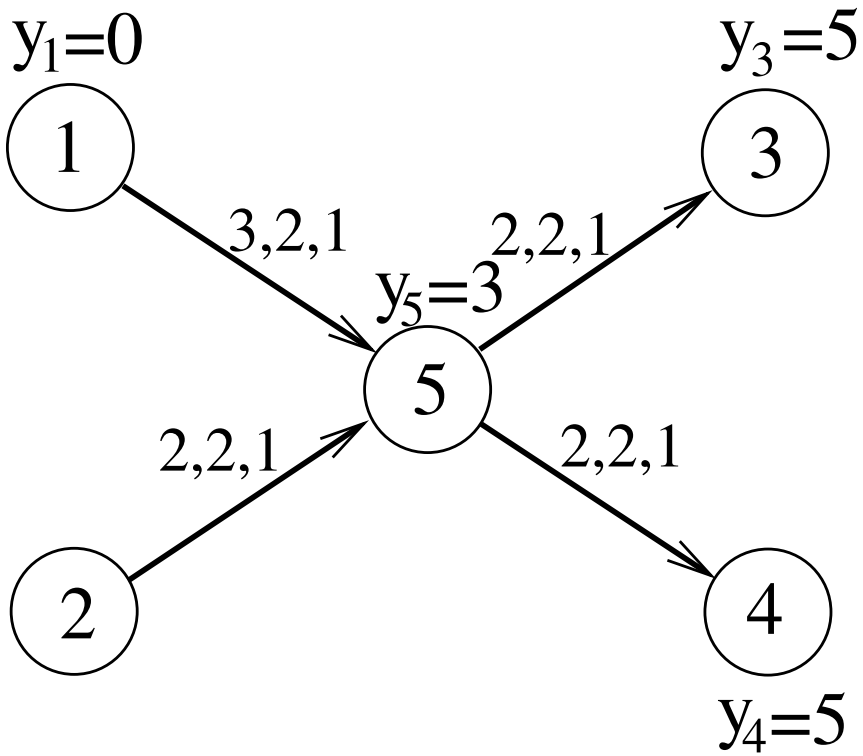
Simplexmetoden för minkostnadsflöde: Exempel

Räkna ut y via basbågarna.



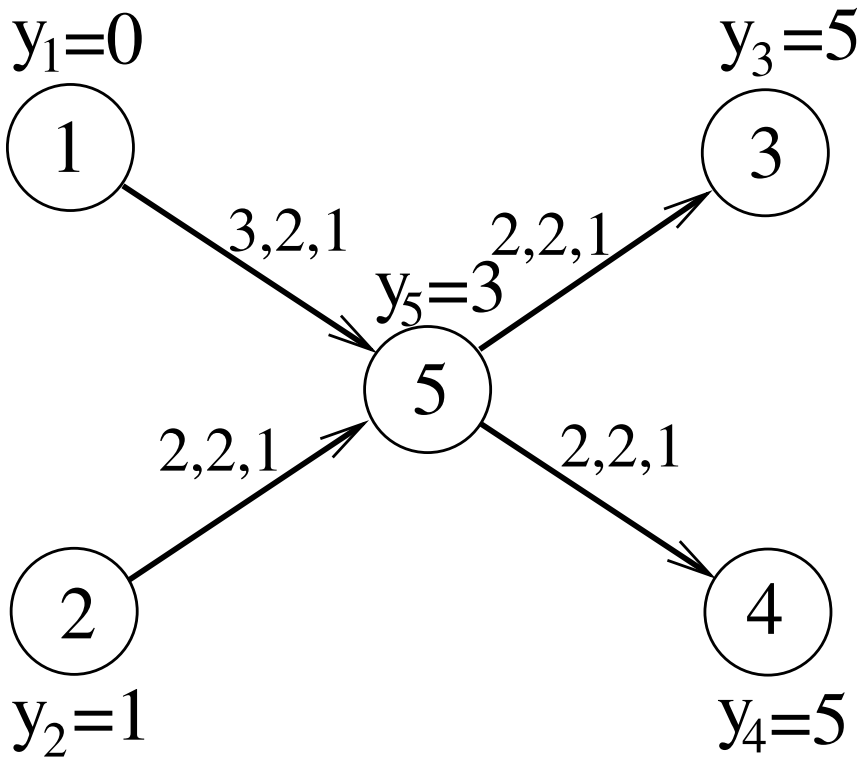
Simplexmetoden för minkostnadsflöde: Exempel

Räkna ut y via basbågarna.



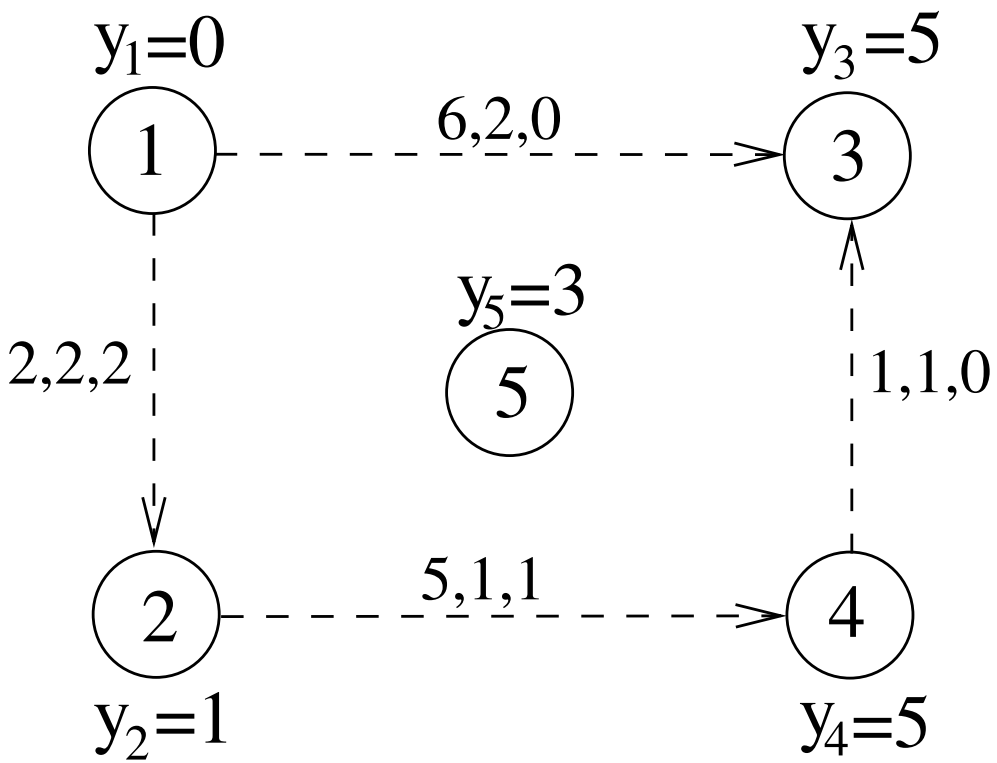
Simplexmetoden för minkostnadsflöde: Exempel

Räkna ut y via basbågarna.



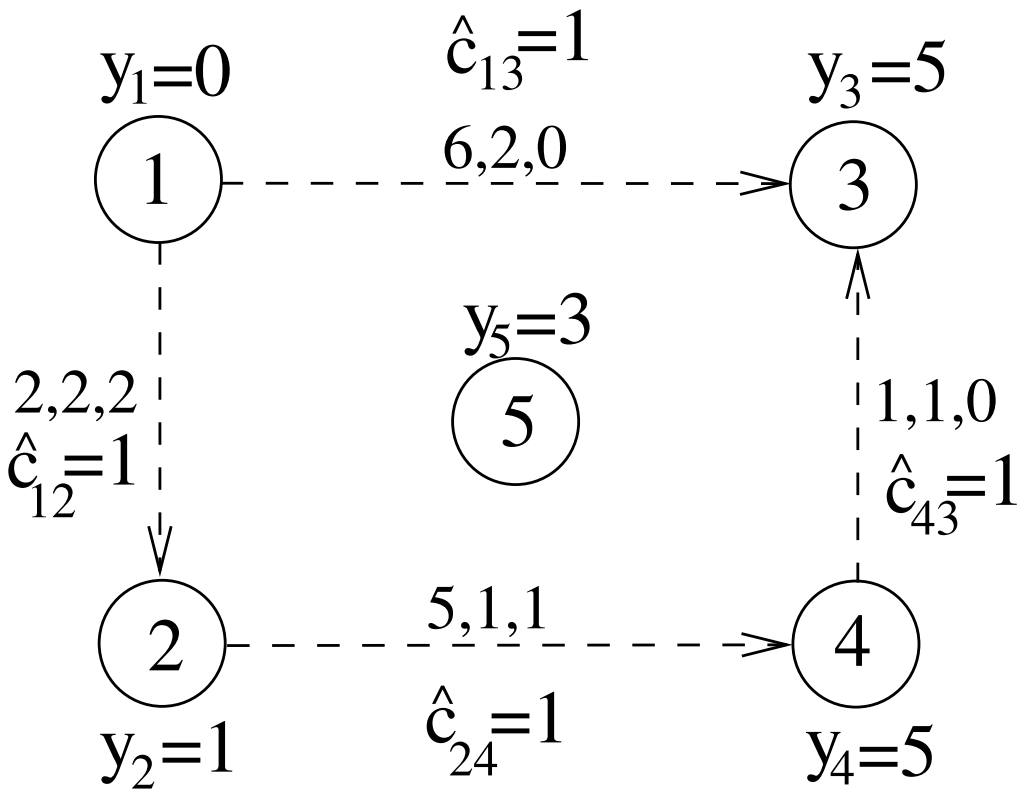
Simplexmetoden för minkostnadsflöde: Exempel

Betrakta icke-basbågarna.



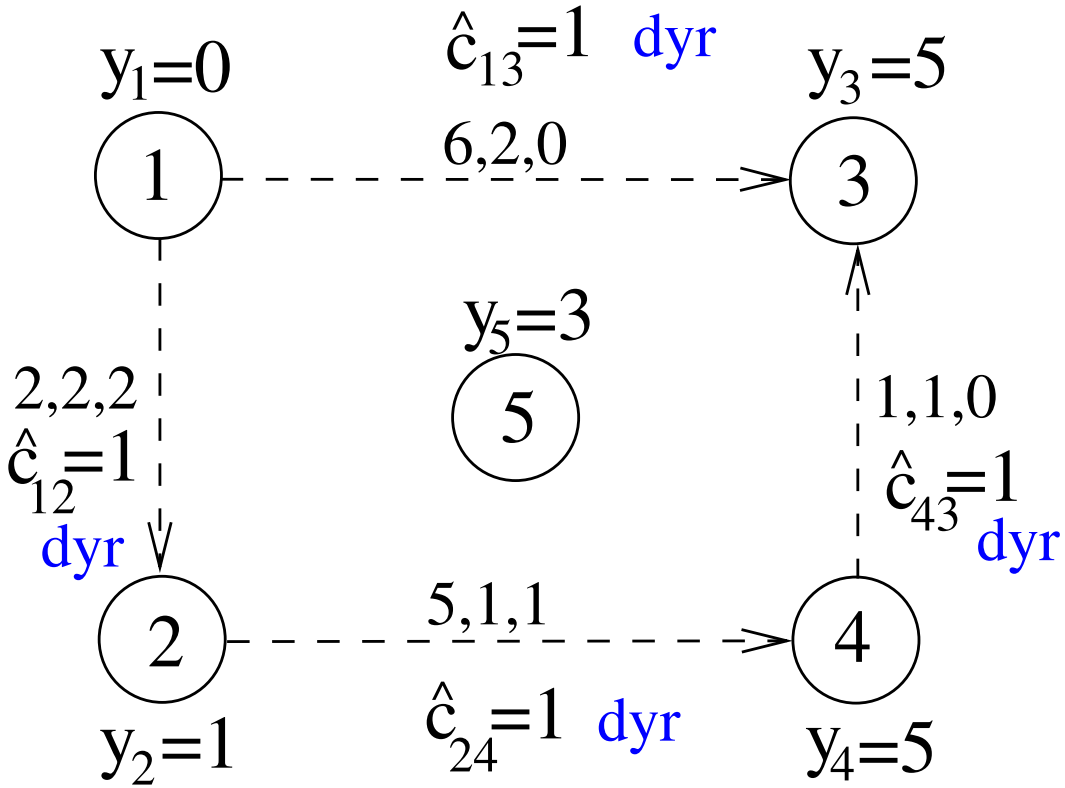
Simplexmetoden för minkostnadsflöde: Exempel

Beräkna reducerade kostnader.



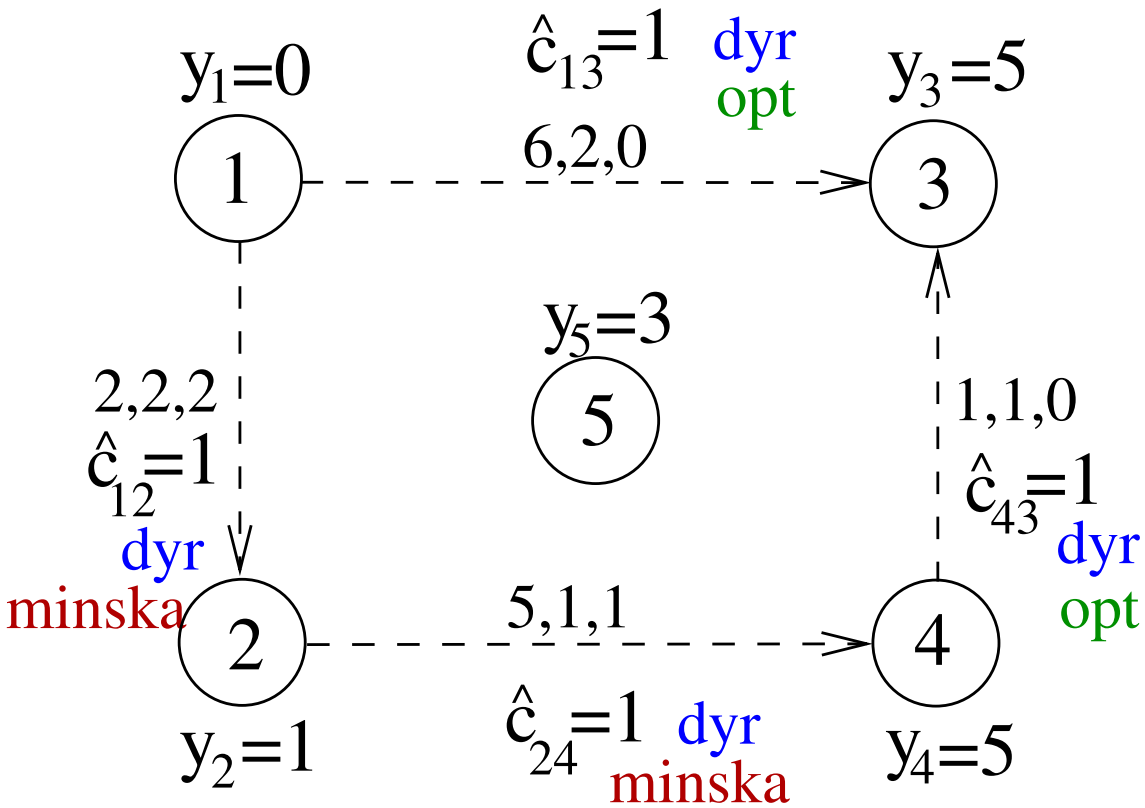
Simplexmetoden för minkostnadsflöde: Exempel

Är flödet optimalt?



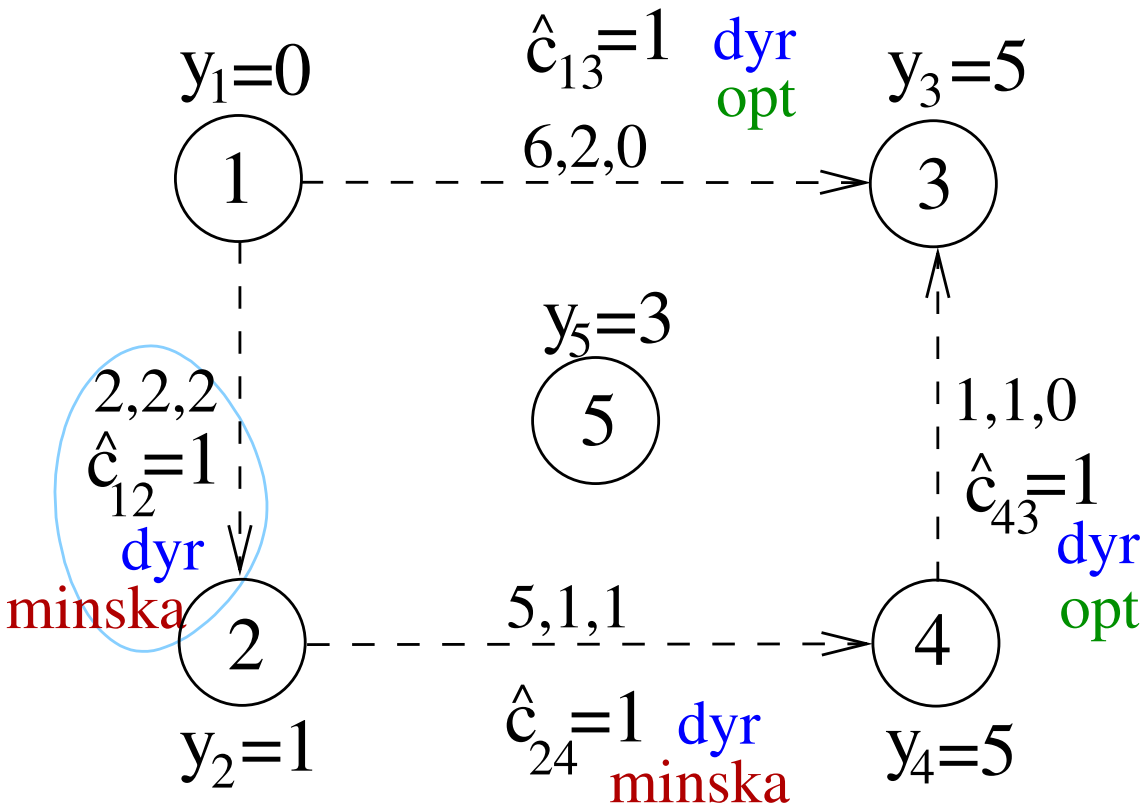
Simplexmetoden för minkostnadsflöde: Exempel

Är flödet optimalt?



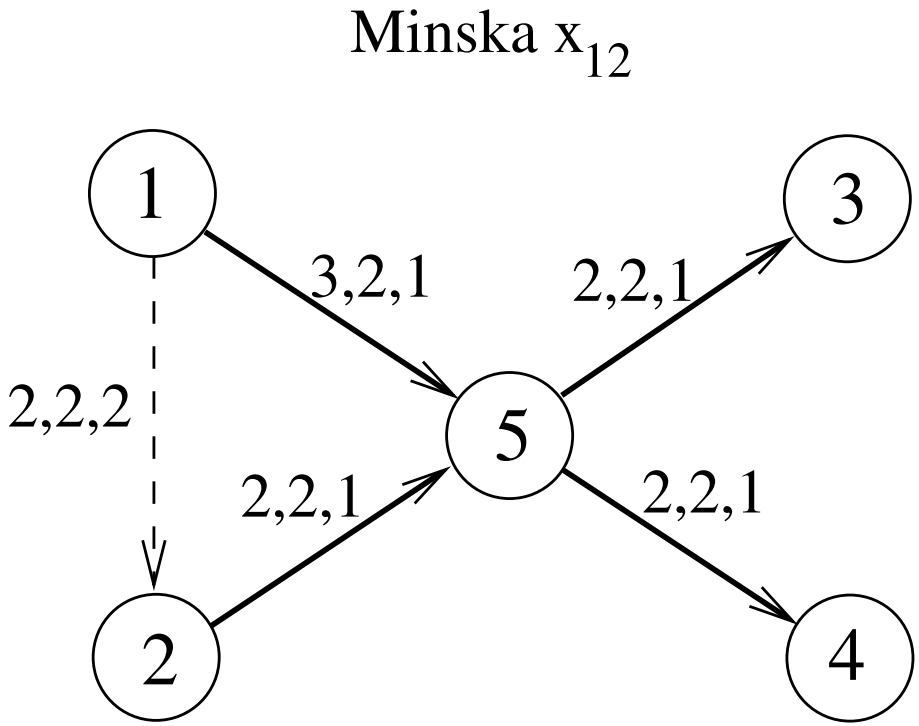
Simplexmetoden för minkostnadsflöde: Exempel

Välj en inkommande variabel.



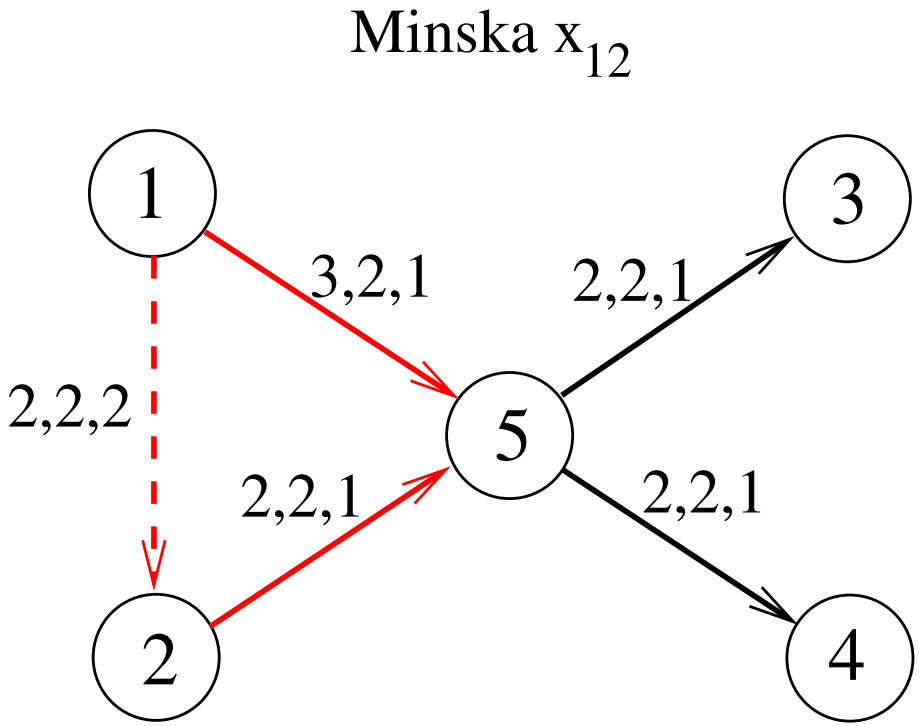
Simplexmetoden för minkostnadsflöde: Exempel

Ändra flödet, via basbågar.



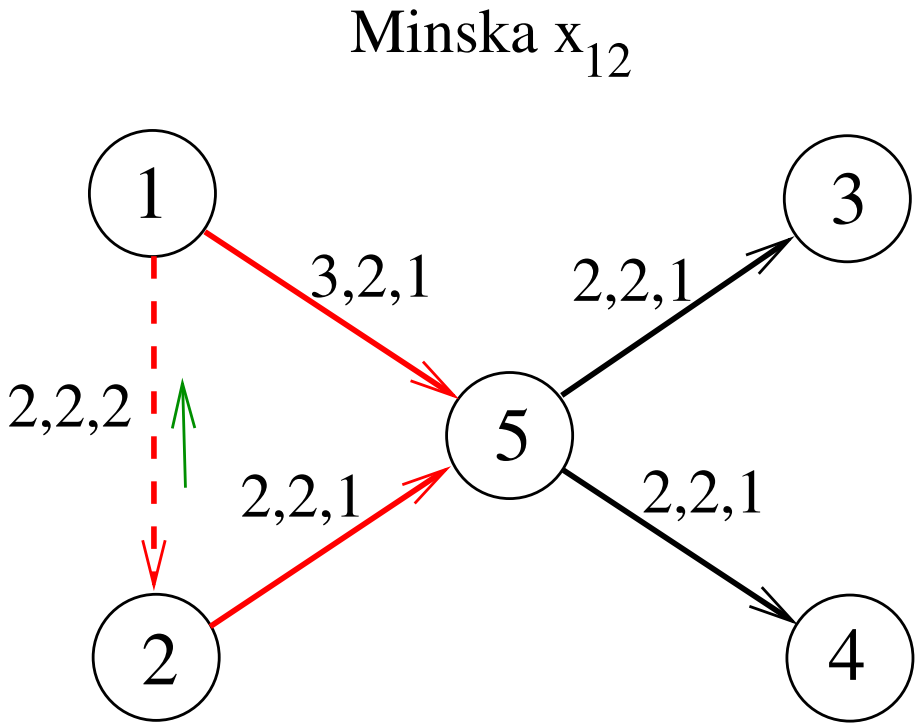
Simplexmetoden för minkostnadsflöde: Exempel

Finn cykel.



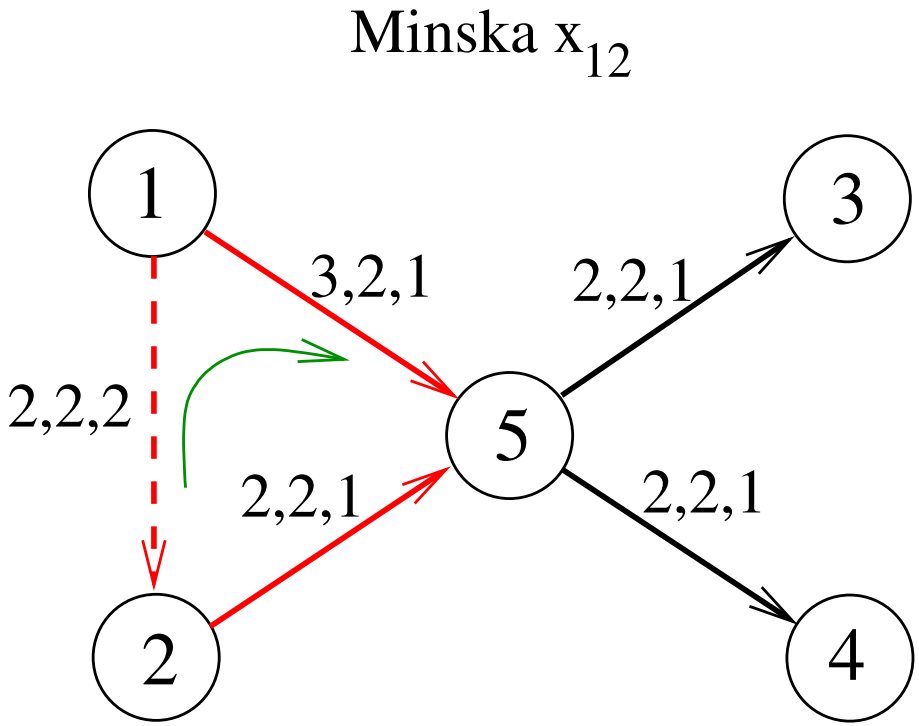
Simplexmetoden för minkostnadsflöde: Exempel

Skicka runt i cykeln.



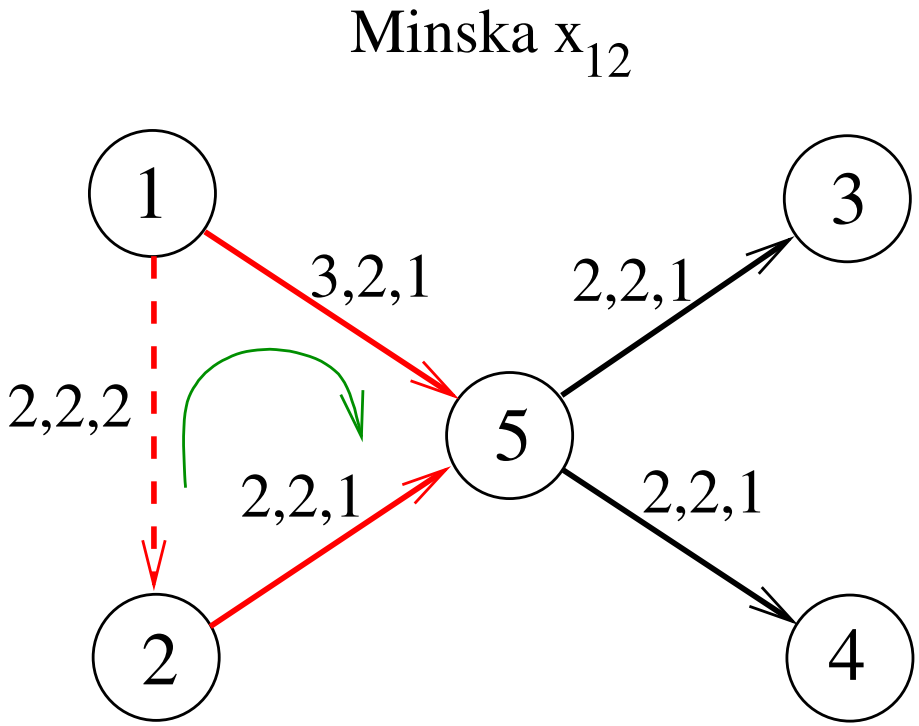
Simplexmetoden för minkostnadsflöde: Exempel

Skicka runt i cykeln.



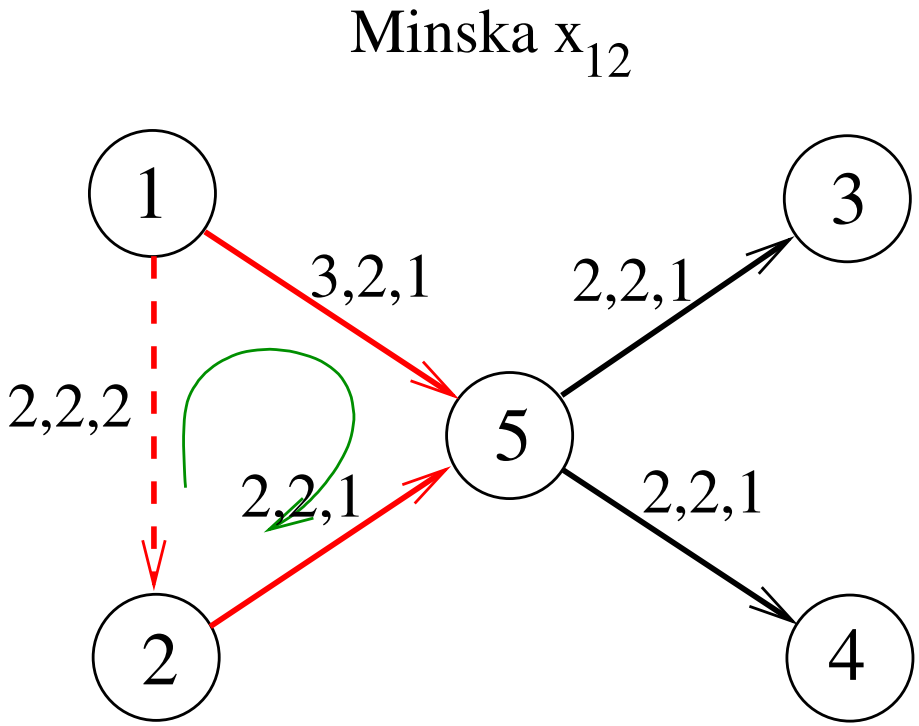
Simplexmetoden för minkostnadsflöde: Exempel

Skicka runt i cykeln.



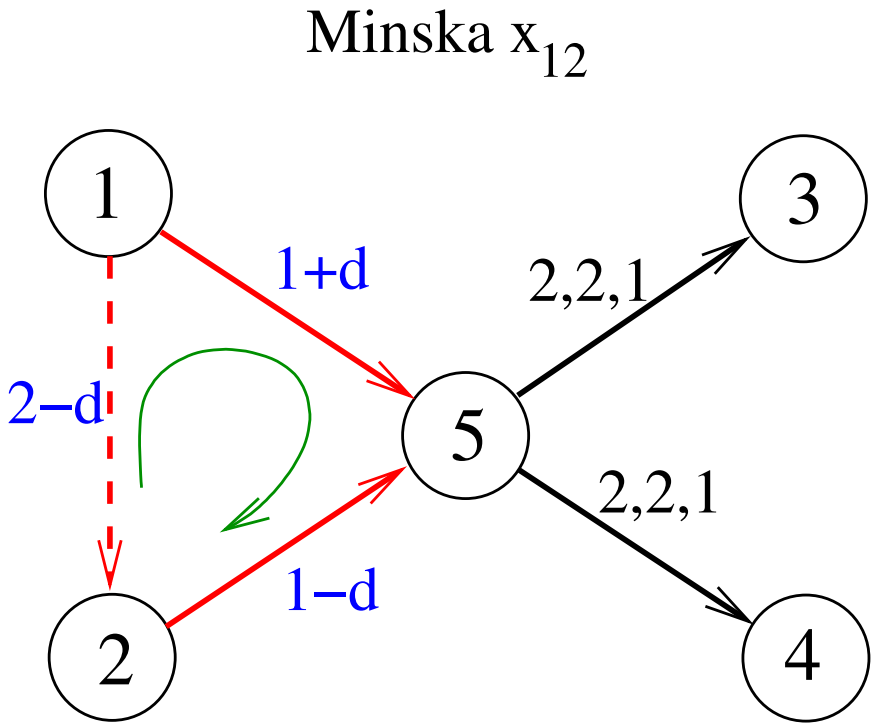
Simplexmetoden för minkostnadsflöde: Exempel

Skicka runt i cykeln.



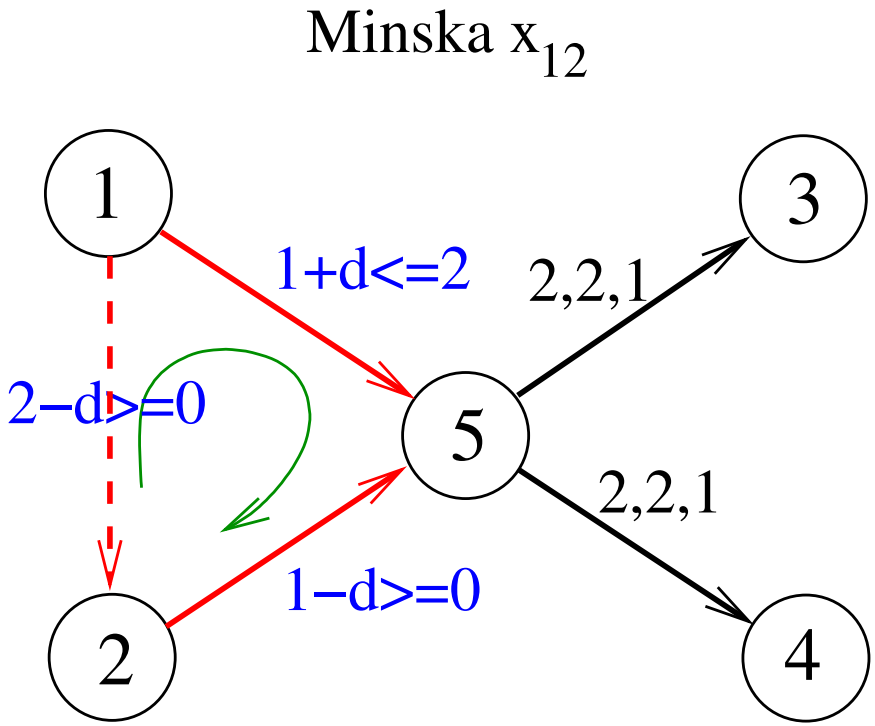
Simplexmetoden för minkostnadsflöde: Exempel

Hur mycket?



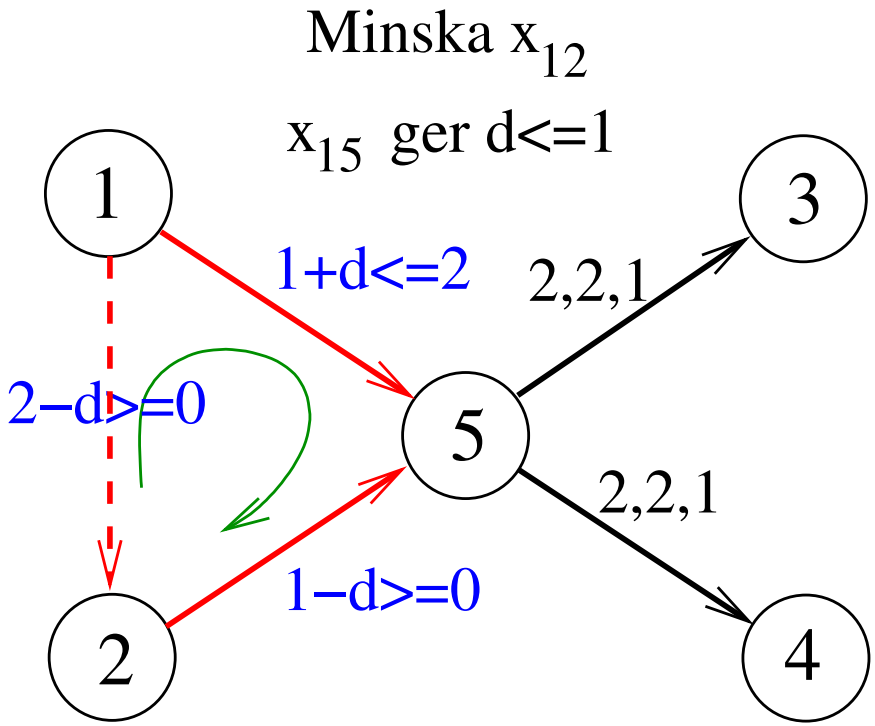
Simplexmetoden för minkostnadsflöde: Exempel

Kolla gränserna.



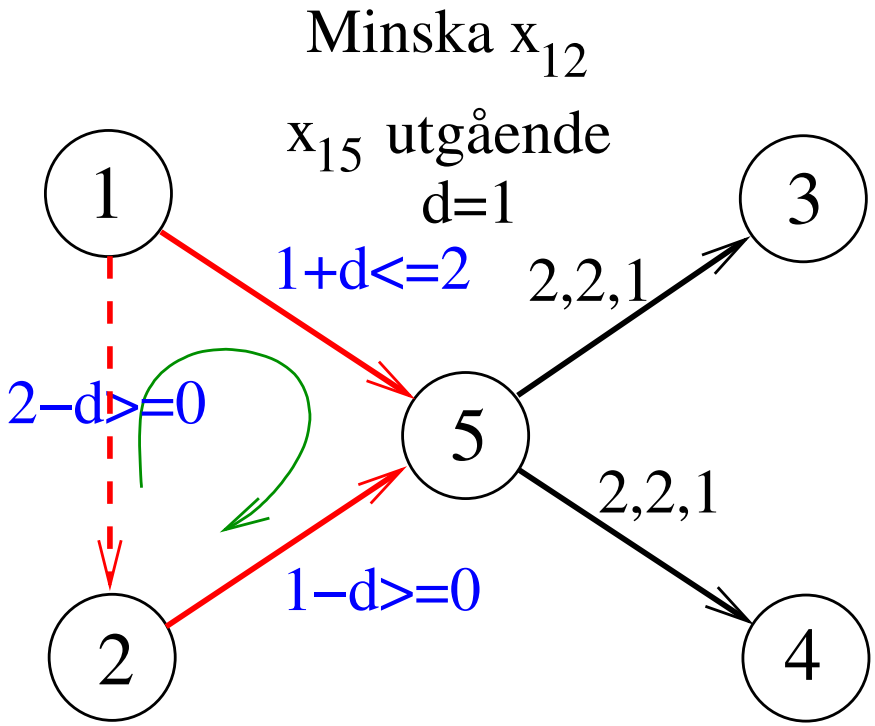
Simplexmetoden för minkostnadsflöde: Exempel

Kolla gränserna.



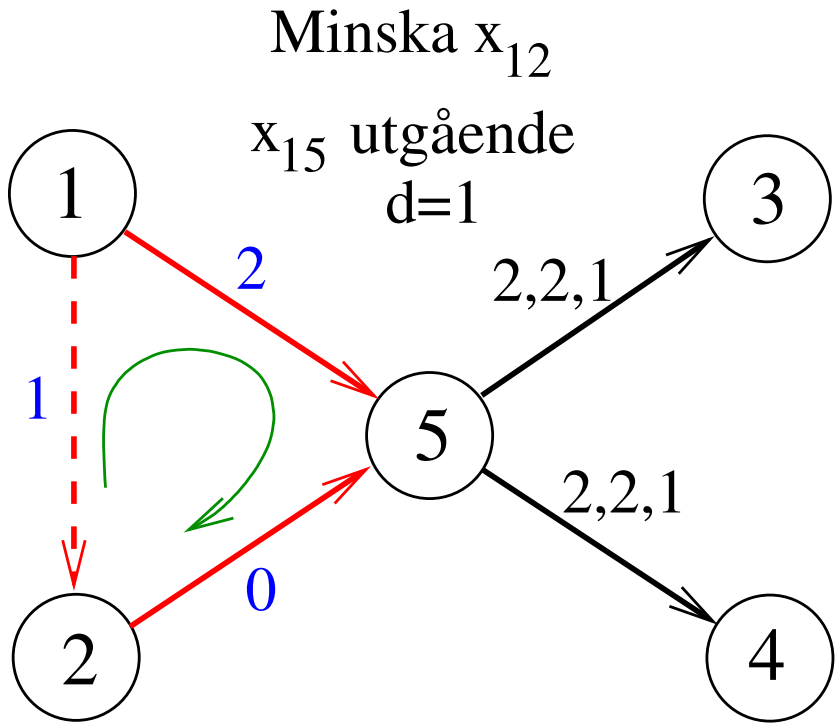
Simplexmetoden för minkostnadsflöde: Exempel

Aktiv begränsning ger utgående variabel.



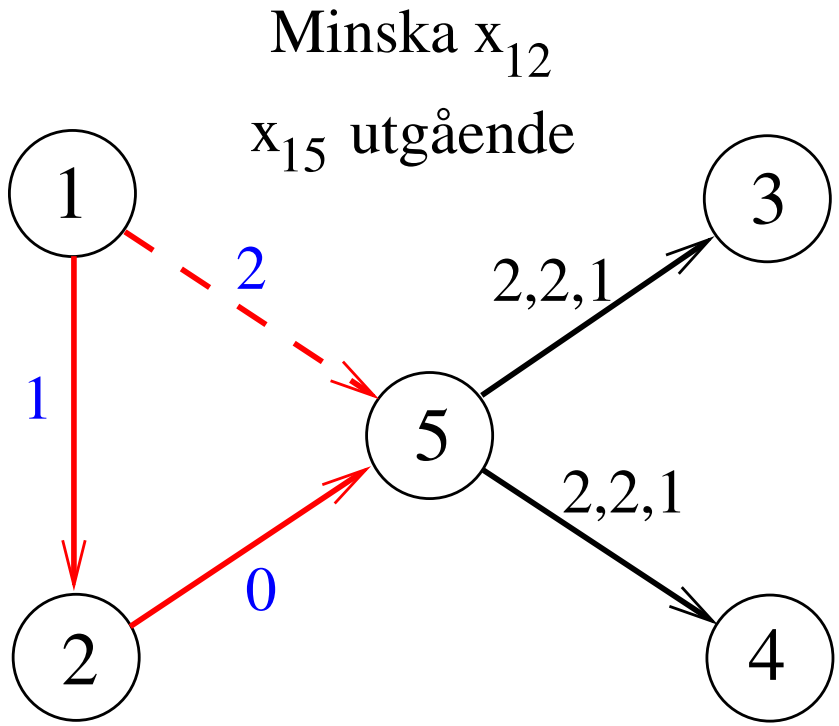
Simplexmetoden för minkostnadsflöde: Exempel

Nytt flöde.



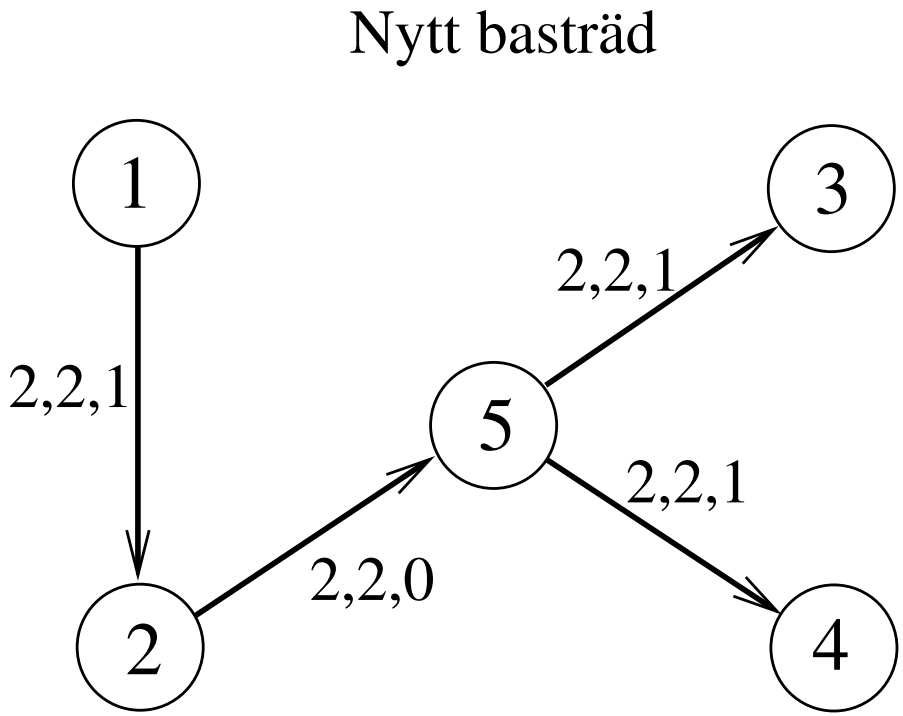
Simplexmetoden för minkostnadsflöde: Exempel

Nytt basträd.



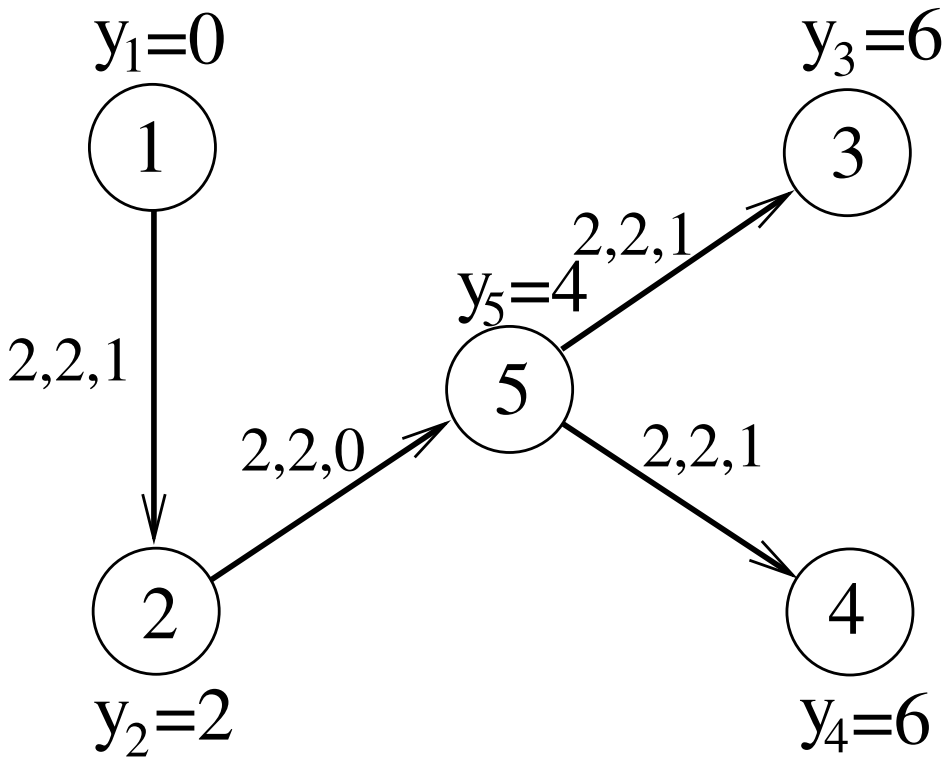
Simplexmetoden för minkostnadsflöde: Exempel

Iteration 2.



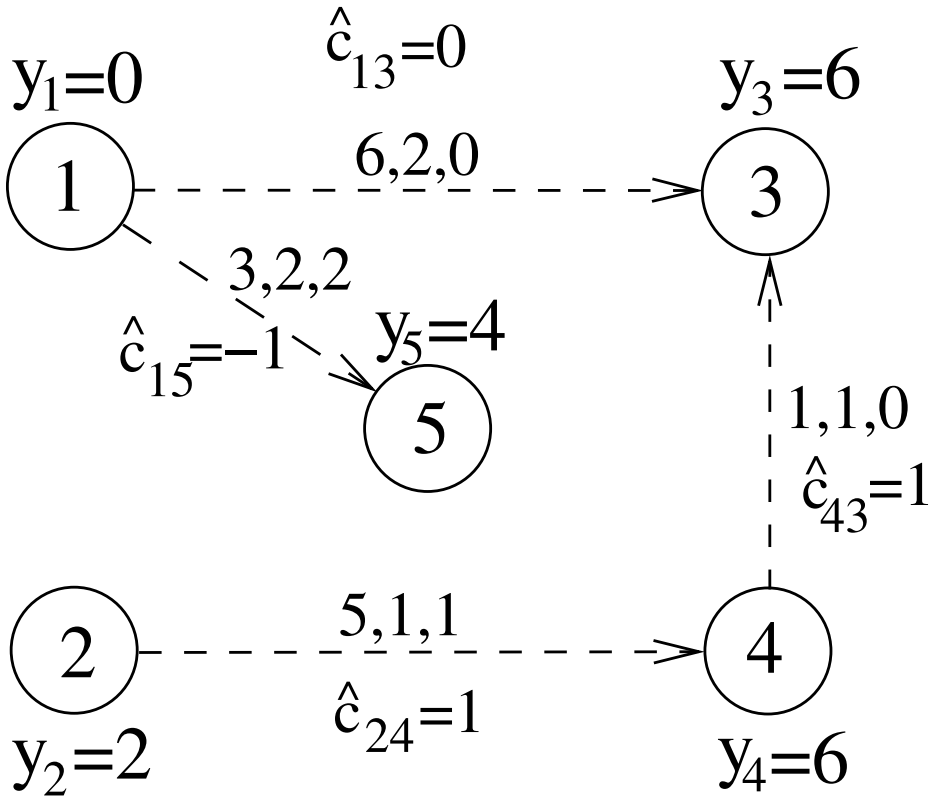
Simplexmetoden för minkostnadsflöde: Exempel

Nodspriser via basträdet.



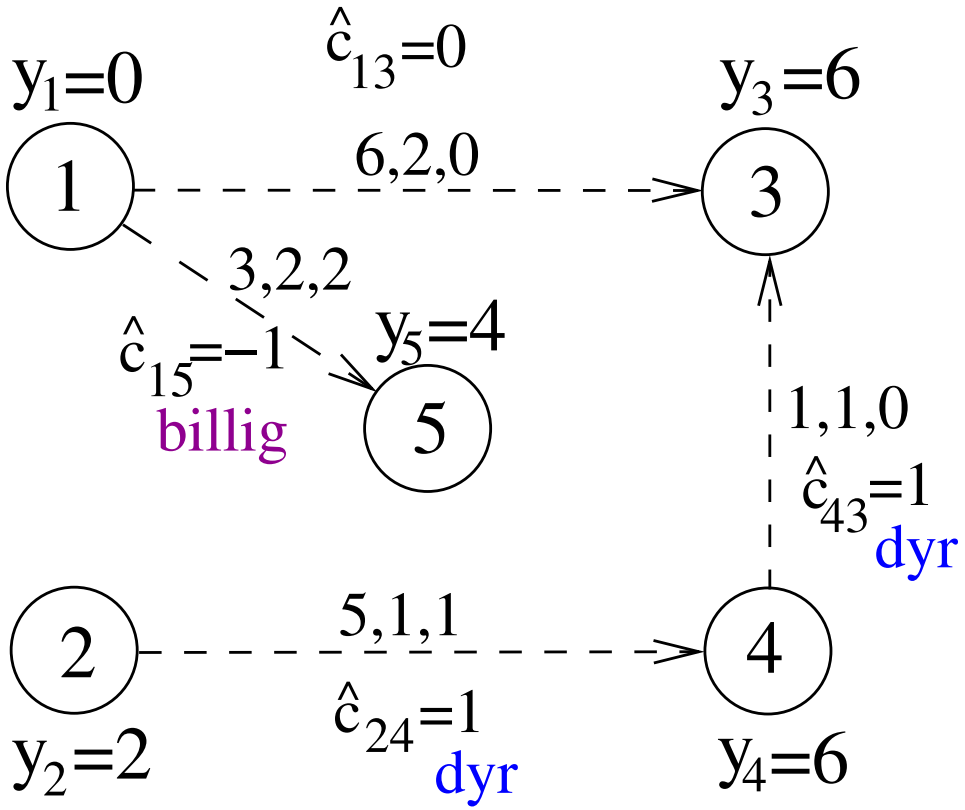
Simplexmetoden för minkostnadsflöde: Exempel

Reducerade kostnader för ickebasbågar.



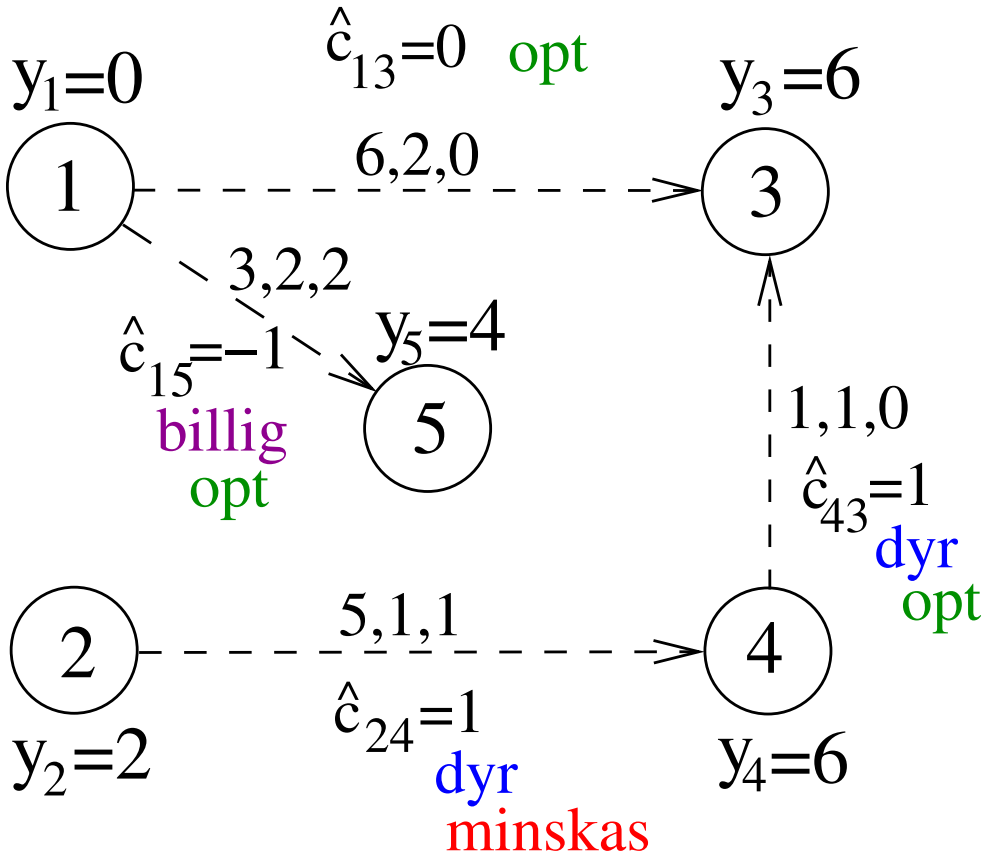
Simplexmetoden för minkostnadsflöde: Exempel

Optimum?



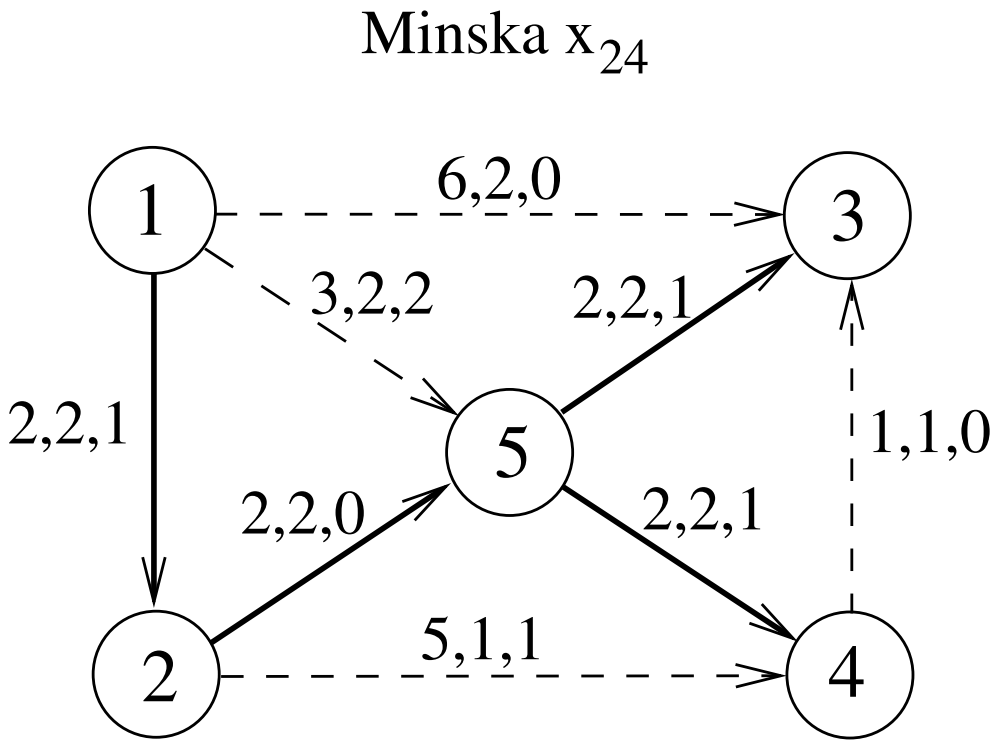
Simplexmetoden för minkostnadsflöde: Exempel

Optimum?



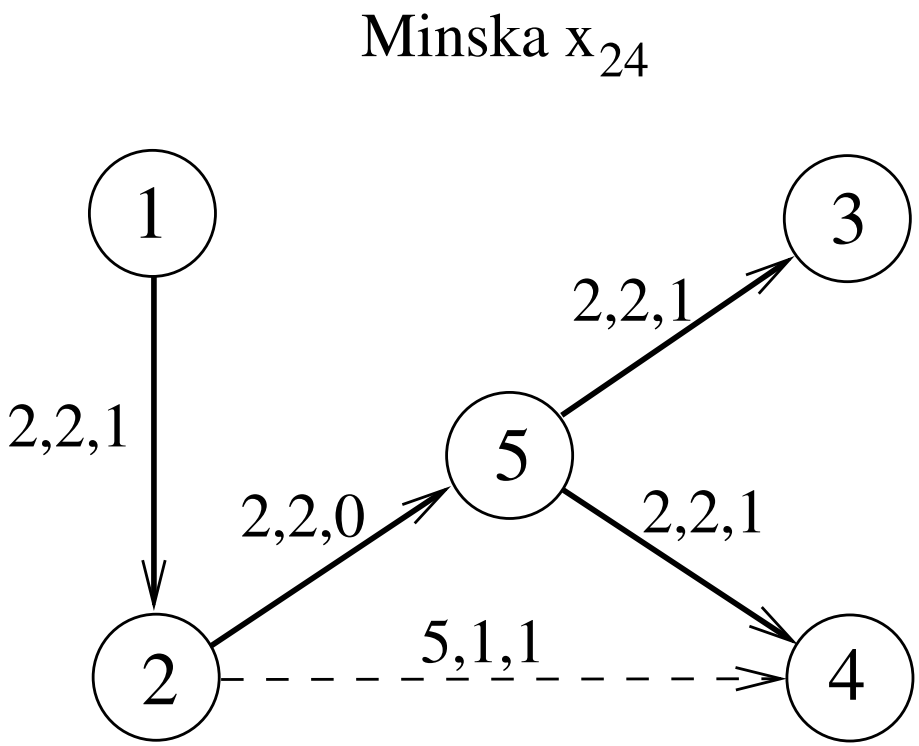
Simplexmetoden för minkostnadsflöde: Exempel

Välj inkommande variabel.



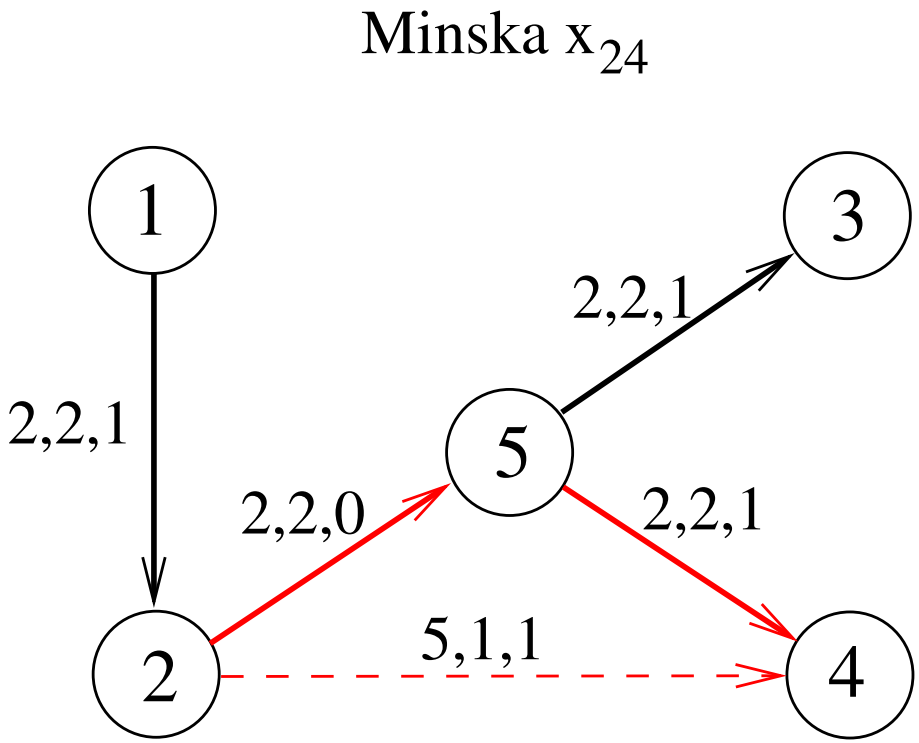
Simplexmetoden för minkostnadsflöde: Exempel

Bortse från andra ickebasbågar.



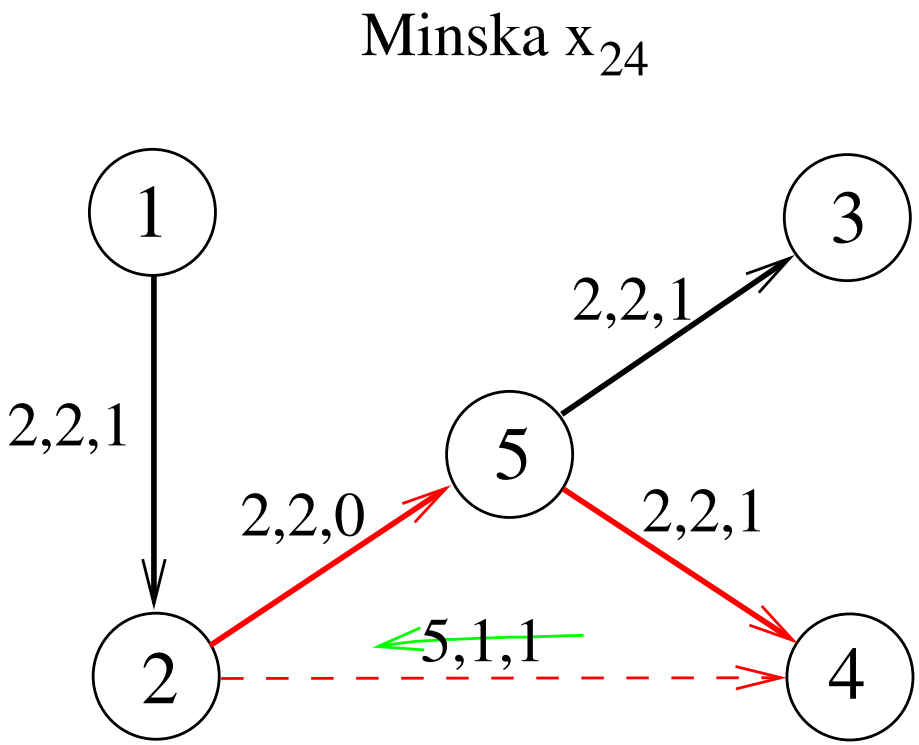
Simplexmetoden för minkostnadsflöde: Exempel

Finn cykeln.



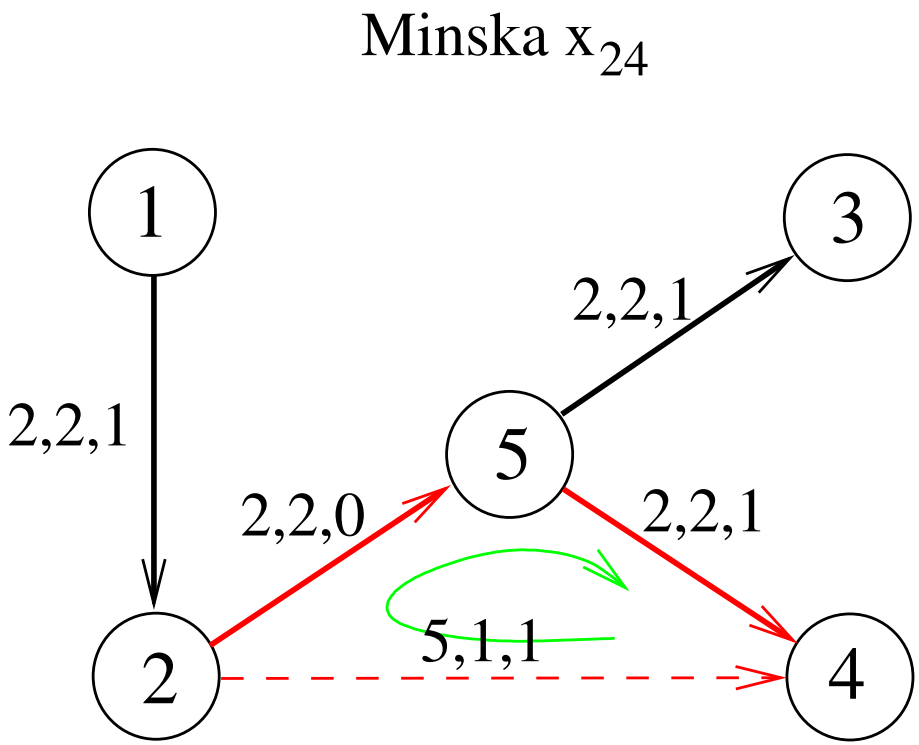
Simplexmetoden för minkostnadsflöde: Exempel

Skicka runt.



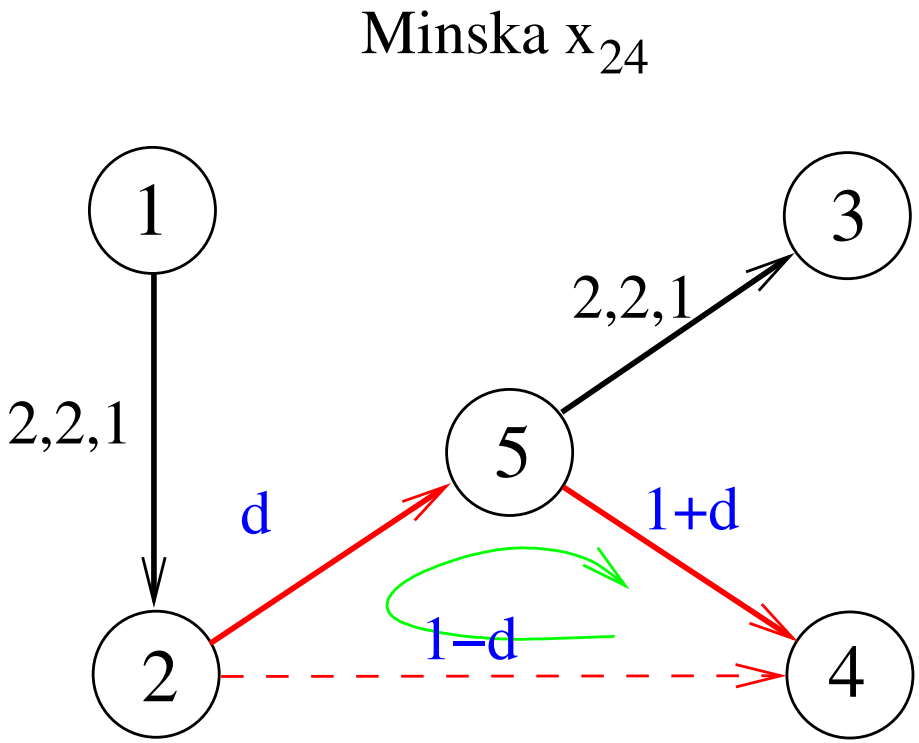
Simplexmetoden för minkostnadsflöde: Exempel

Skicka runt.



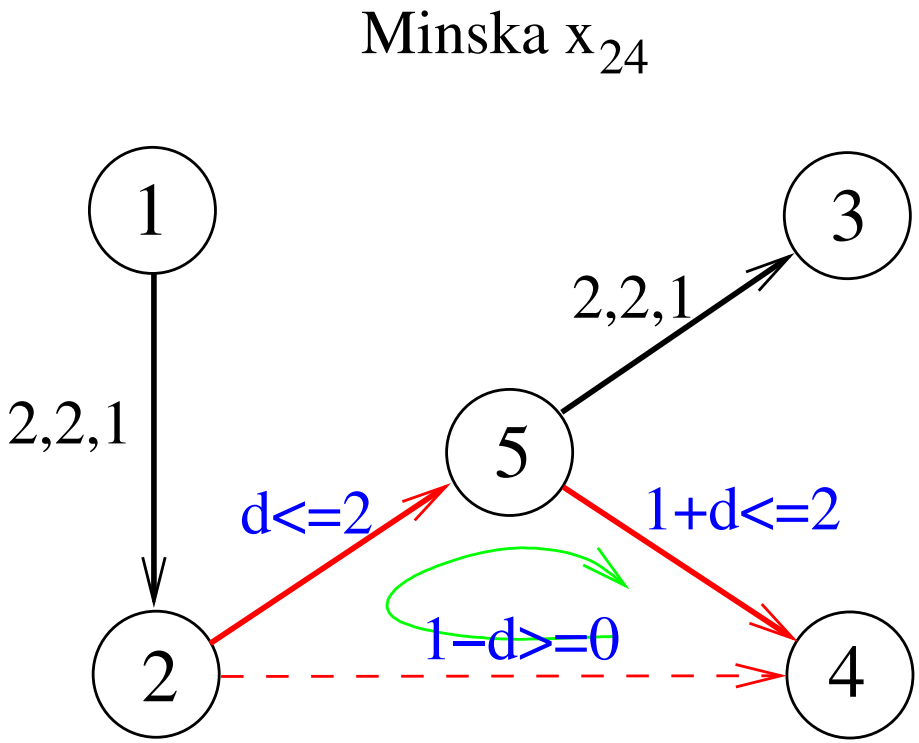
Simplexmetoden för minkostnadsflöde: Exempel

Hur mycket?



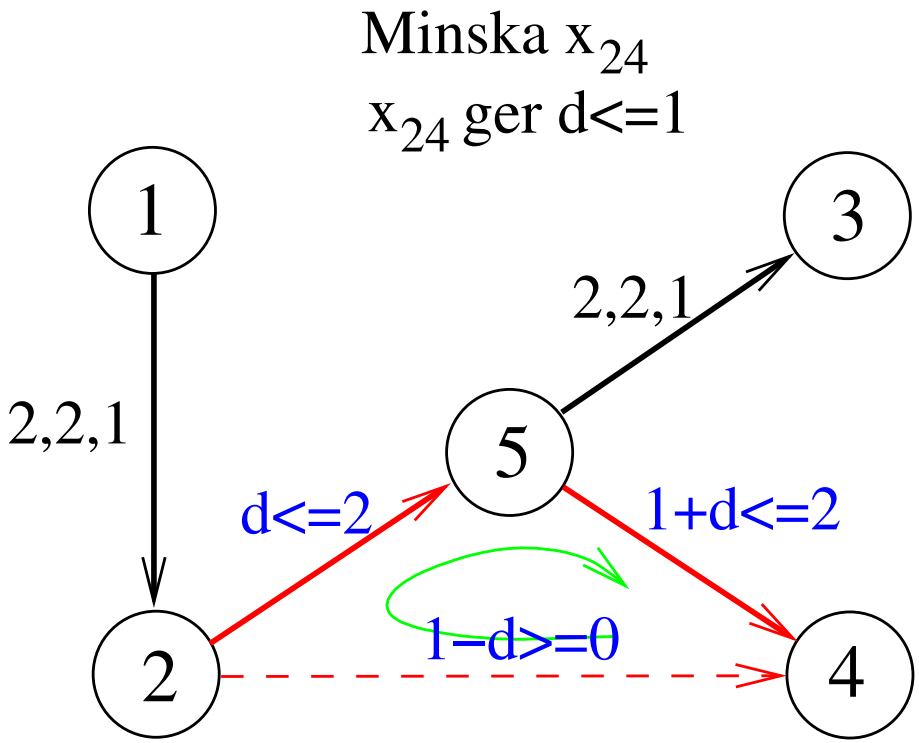
Simplexmetoden för minkostnadsflöde: Exempel

Kolla gränser.



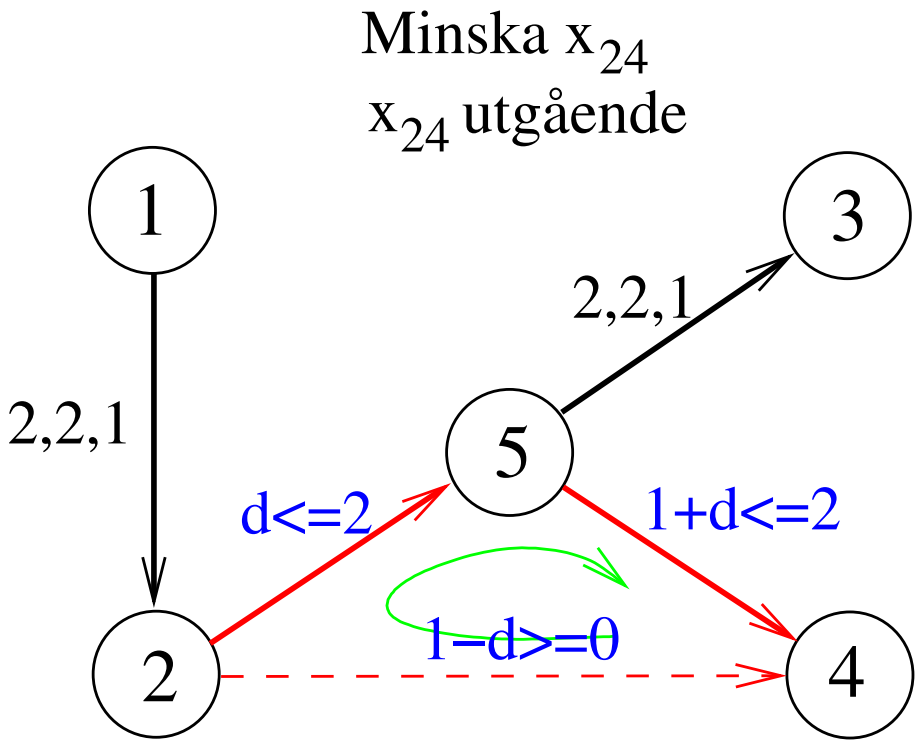
Simplexmetoden för minkostnadsflöde: Exempel

Finn utgående variabel.



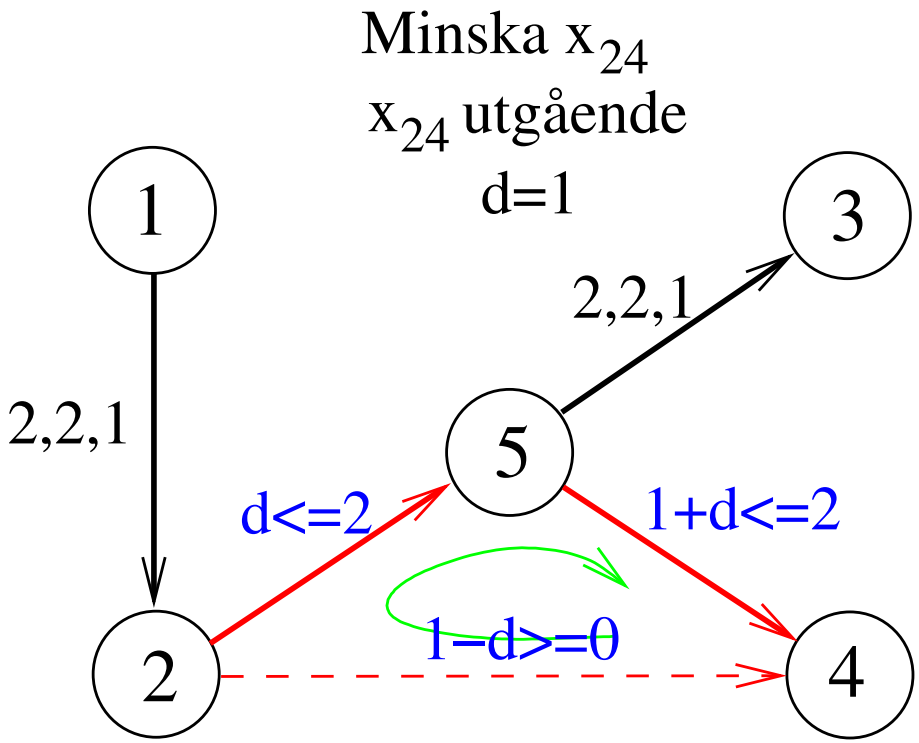
Simplexmetoden för minkostnadsflöde: Exempel

Finn utgående variabel.



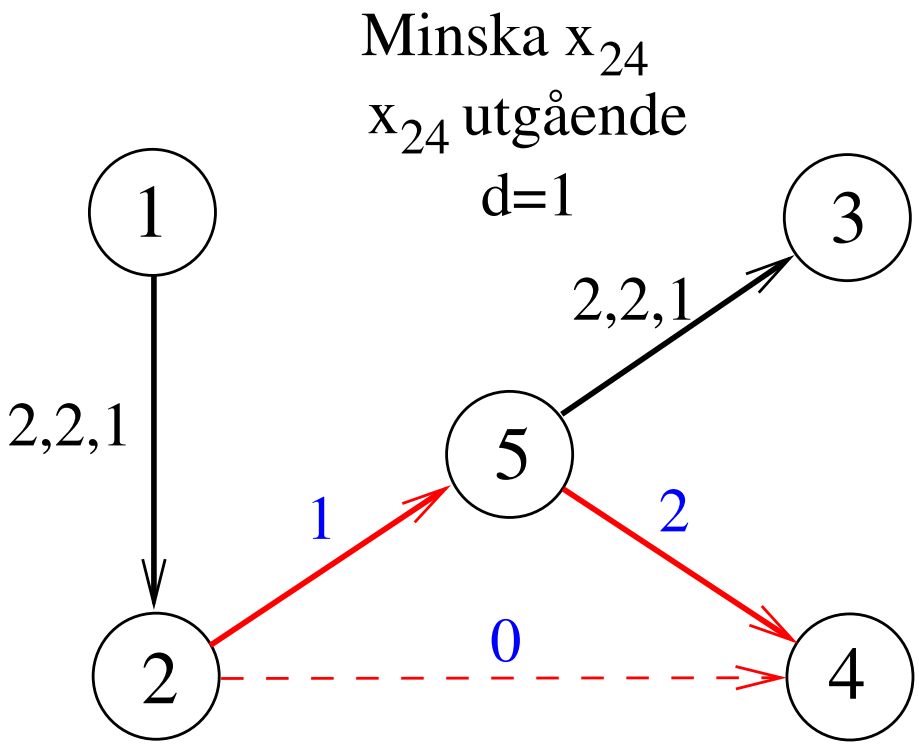
Simplexmetoden för minkostnadsflöde: Exempel

Ändra flödet.



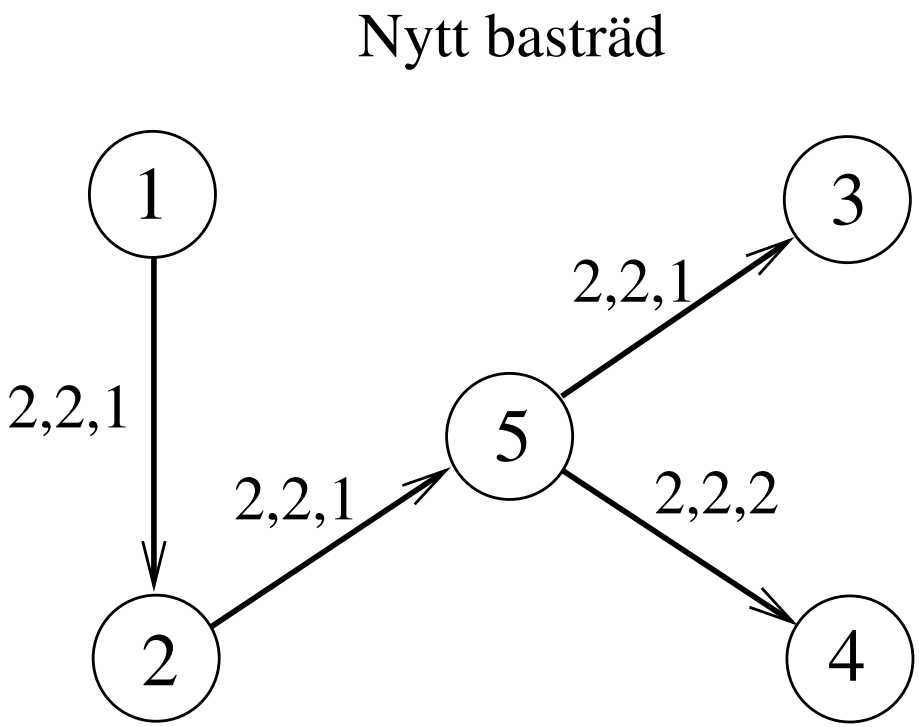
Simplexmetoden för minkostnadsflöde: Exempel

Ändra flödet.



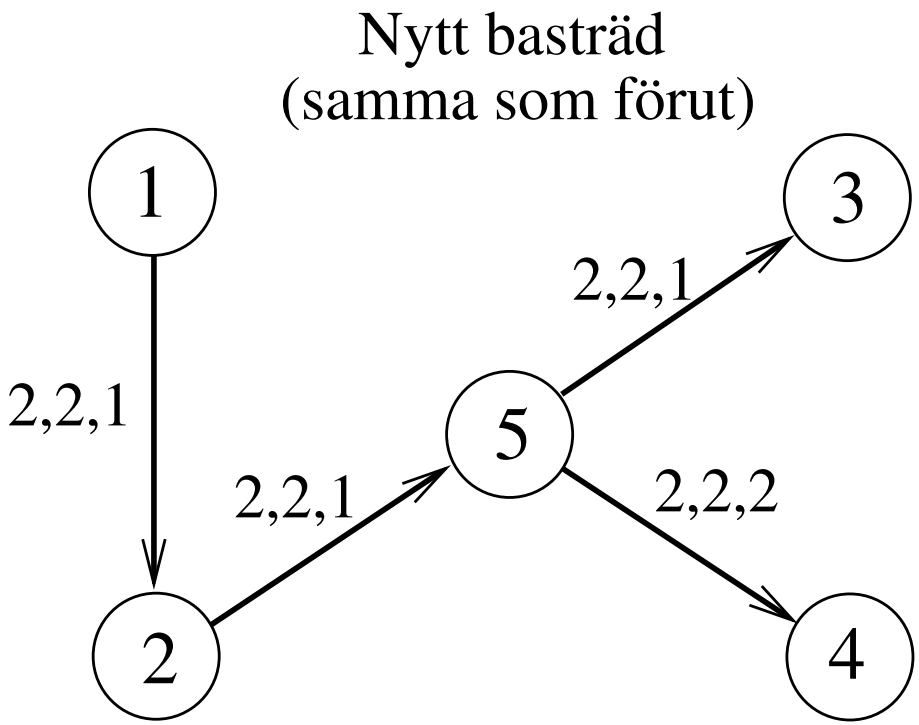
Simplexmetoden för minkostnadsflöde: Exempel

Iteration 3.



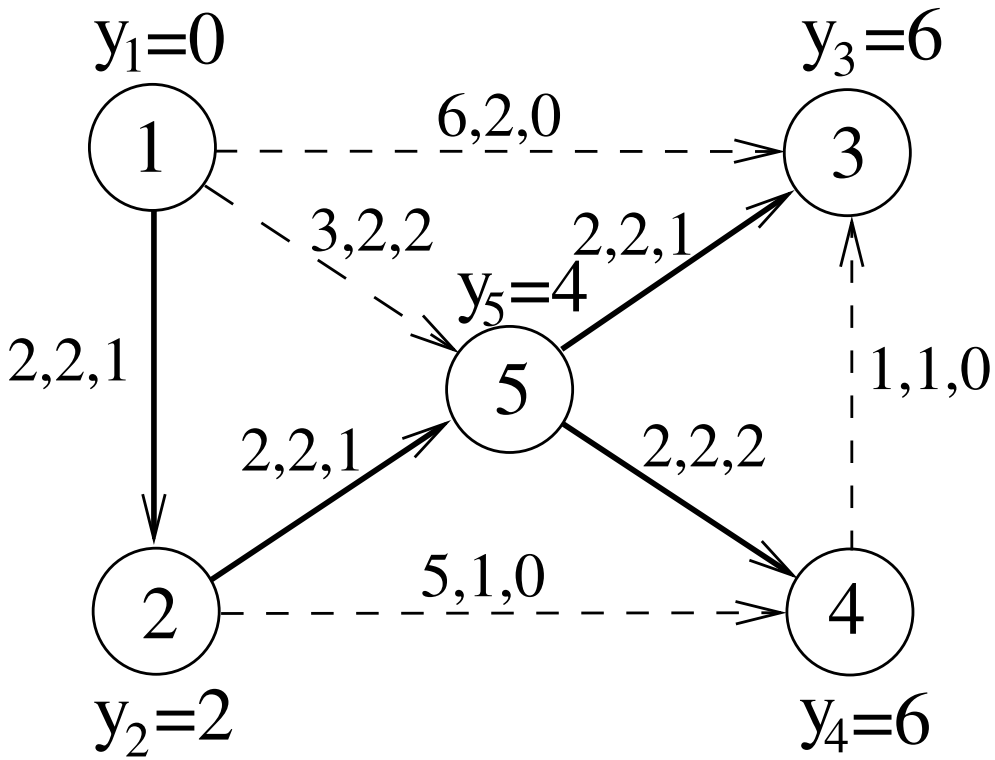
Simplexmetoden för minkostnadsflöde: Exempel

Iteration 3.



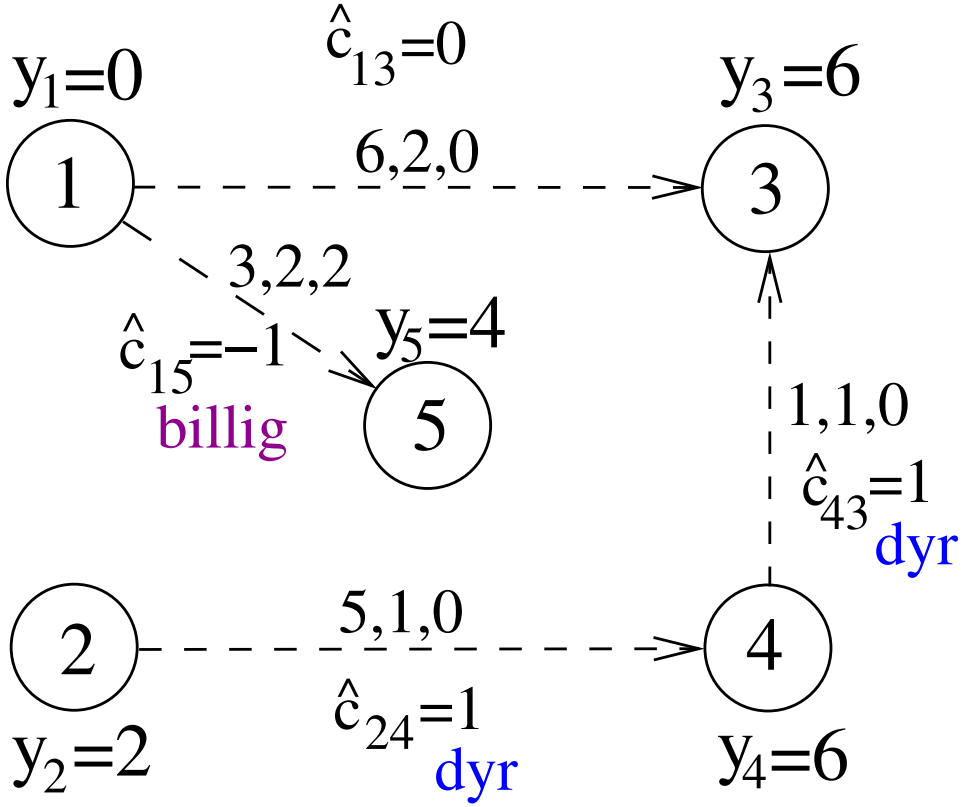
Simplexmetoden för minkostnadsflöde: Exempel

Nodpriser.



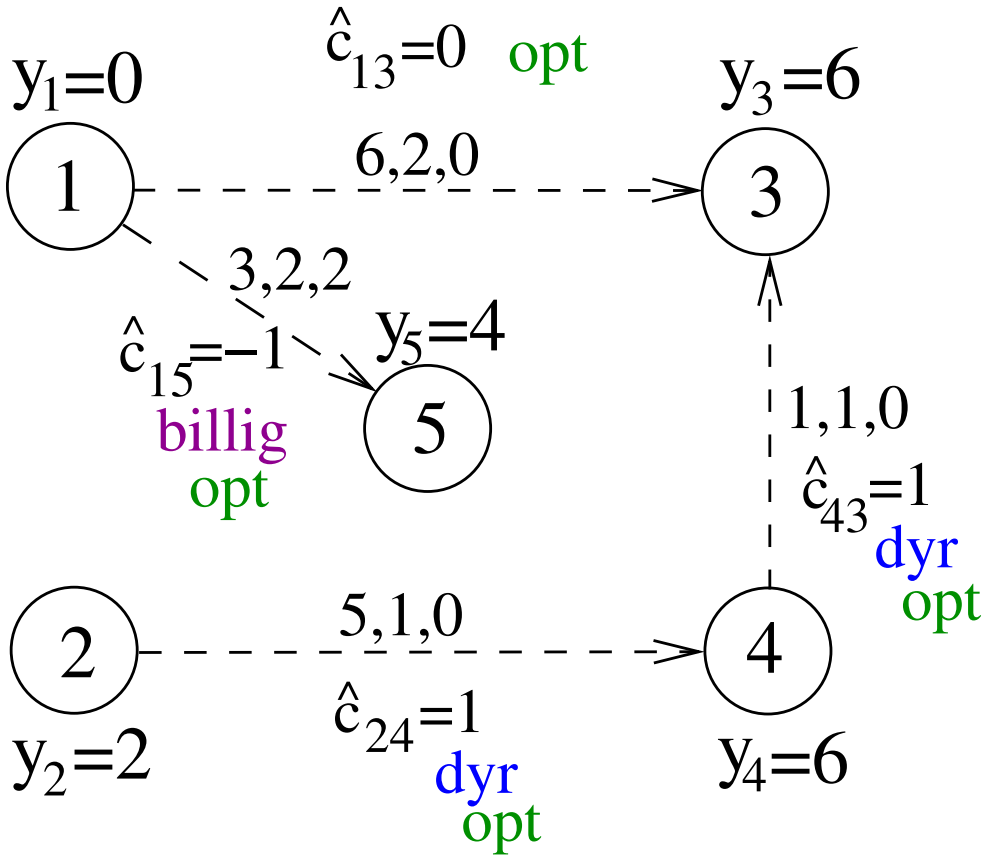
Simplexmetoden för minkostnadsflöde: Exempel

Reducerade kostnader.



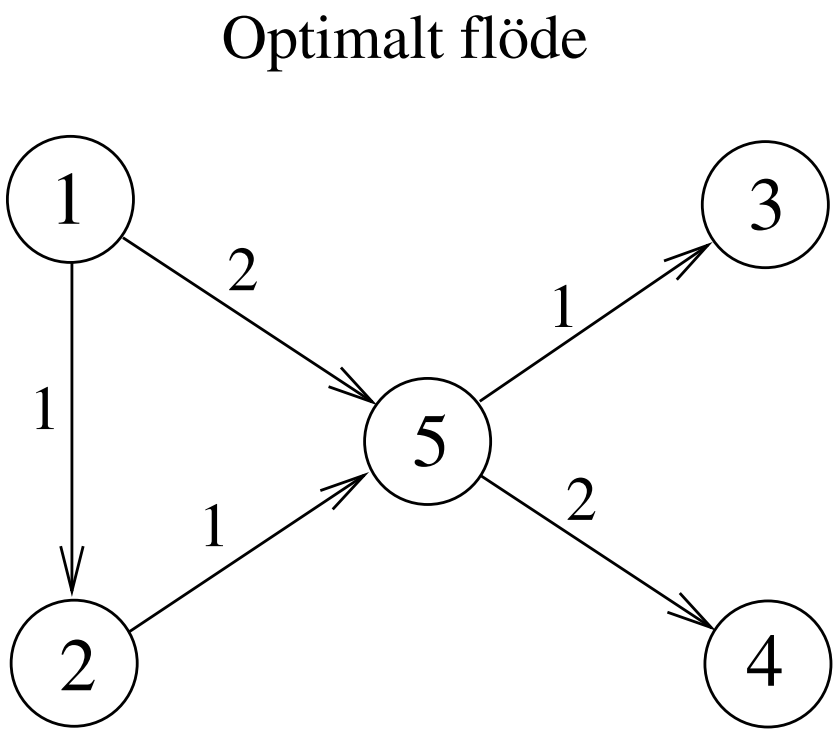
Simplexmetoden för minkostnadsflöde: Exempel

Optimum?



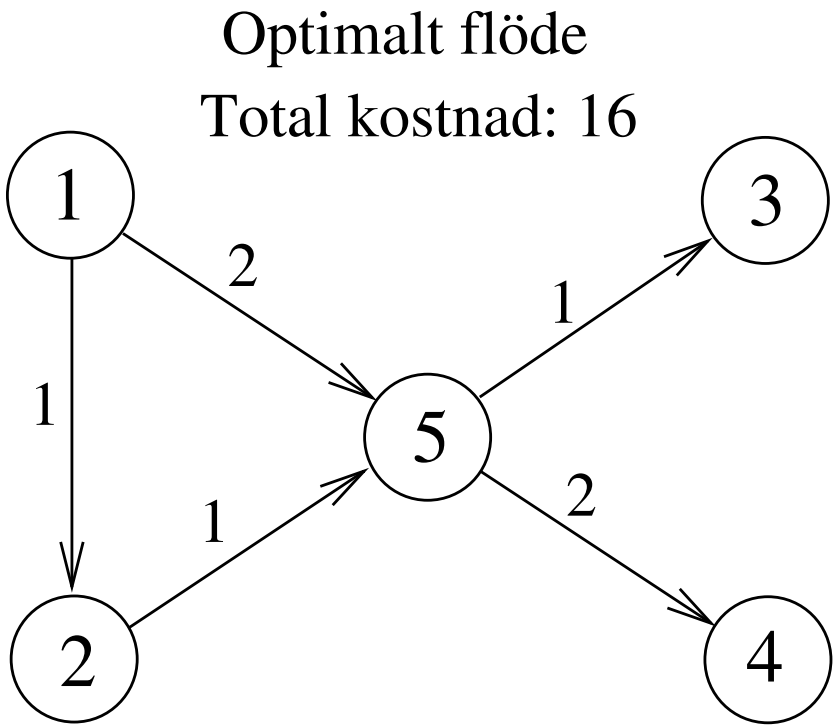
Simplexmetoden för minkostnadsflöde: Exempel

Ja, optimum!



Simplexmetoden för minkostnadsflöde: Exempel

Beräkna totalkostnaden separat: $\sum_{ij} c_{ij} x_{ij}$.



Simplexmetoden för minkostnadsflödesproblemet

Känslighetsanalys

Simplexmetoden för minkostnadsflödesproblemet

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.

Busacker-Gowens metod

Alternativ metod för minkostnadsflödesproblem med en källa och en sänka:

Busacker-Gowens metod

Alternativ metod för minkostnadsflödesproblem med en källa och en sänka:

Modifiering av maxflödesmetoden:

Busacker-Gowens metod

Alternativ metod för minkostnadsflödesproblem med en källa och en sänka:

Modifiering av maxflödesmetoden:

Sök *billigaste* flödesökande väg i varje iteration.

Busacker-Gowens metod

Alternativ metod för minkostnadsflödesproblem med en källa och en sänka:

Modifiering av maxflödesmetoden:

Sök *billigaste* flödesökande väg i varje iteration.

Pseudopolynomisk.

Busacker-Gowens metod

Alternativ metod för minkostnadsflödesproblem med en källa och en sänka:

Modifiering av maxflödesmetoden:

Sök *billigaste* flödesökande väg i varje iteration.

Pseudopolynomisk.

Billigaste-maxflödesmetoden

Busacker-Gowens metod

Alternativ metod för minkostnadsflödesproblem med en källa och en sänka:

Modifiering av maxflödesmetoden:

Sök *billigaste* flödesökande väg i varje iteration.

Pseudopolynomisk.

Billigaste-maxflödesmetoden

Busacker-Gowens metod kan finna billigaste maxflöde om man inte slutar förrän flödet är maximalt.

Tillordningsproblemet

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

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.

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.

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

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.

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

Metod: Formulera och lös LP-dualen.

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.

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

Metod: Formulera och lös LP-dualen.

Arbeta med dualvariabler och reducerade kostnader.

Tillordningsproblemet: Lös LP-dualen

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

Tillordningsproblemet: Lös LP-dualen

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

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

Tillordningsproblemet: Lös LP-dualen

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

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

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

Tillordningsproblemet: Lös LP-dualen

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

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

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

Metod: Lös dualen:

Tillordningsproblemet: Lös LP-dualen

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

LP-dual: $\max v = \sum_{i=1}^{n} \alpha_i + \sum_{j=1}^{n} \beta_j$ då $\alpha_i + \beta_j \leq c_{ij}$ 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,

Tillordningsproblemet: Lös LP-dualen

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

LP-dual: $\max v = \sum_{i=1}^{n} \alpha_i + \sum_{j=1}^{n} \beta_j$ då $\alpha_i + \beta_j \leq c_{ij}$ 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.

Tillordningsproblemet: Lös LP-dualen

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

LP-dual: $\max v = \sum_{i=1}^{n} \alpha_i + \sum_{j=1}^{n} \beta_j$ då $\alpha_i + \beta_j \leq c_{ij}$ 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.

Tillordningsproblemet: Lös LP-dualen

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

LP-dual: $\max v = \sum_{i=1}^{n} \alpha_i + \sum_{j=1}^{n} \beta_j$ då $\alpha_i + \beta_j \leq c_{ij}$ 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

Arbeta med reducerade kostnader:

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

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)

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)

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

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 $\hat{c}_{ij} = 0$.

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.

Ungerska metoden för tillordningsproblemet

Lös dualen.

Ungerska metoden för tillordningsproblemet

Lös dualen.

- Sätt $\alpha_i = 0$ för alla i och $\beta_j = 0$ för alla j (vilket ger $\hat{c}_{ij} = c_{ij}$).

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).

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.

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.

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,

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.

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, $\hat{C}$.

Ungerska metoden för tillordningsproblemet

Matris av reducerade kostnader, $\hat{C}$.

Stryk alla tillåtna positioner med minsta möjliga antal streck.

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.

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.

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.

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.

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.

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 för tillordningsproblemet

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

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).

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).

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.

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.

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

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

					α
	3	4	2	5	0
	7	8	5	7	0
	2	3	2	3	0
	5	8	9	7	0
β	0	0	0	0	

Starta med α och β noll.

Ungerska metoden: Exempel

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

					α
	3	4	2	5	0
	7	8	5	7	0
	2	3	2	3	0
	5	8	9	7	0
β	0	0	0	0	

Finn minsta element i rad 1.

Ungerska metoden: Exempel

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

					α
	1	2	0	3	2
	7	8	5	7	0
	2	3	2	3	0
	5	8	9	7	0
β	0	0	0	0	

Öka α_1 så mycket, vilket minskar rad 1.

Ungerska metoden: Exempel

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

					α
	1	2	0	3	2
	7	8	5	7	0
	2	3	2	3	0
	5	8	9	7	0
β	0	0	0	0	

Finn minsta element i rad 2.

Ungerska metoden: Exempel

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

					α
	1	2	0	3	2
	2	3	0	2	5
	2	3	2	3	0
	5	8	9	7	0
β	0	0	0	0	

Öka α_2 så mycket, vilket minskar rad 2.

Ungerska metoden: Exempel

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

					α
	1	2	0	3	2
	2	3	0	2	5
	2	3	2	3	0
	5	8	9	7	0
β	0	0	0	0	

Finn minsta element i rad 3.

Ungerska metoden: Exempel

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

					α
	1	2	0	3	2
	2	3	0	2	5
	0	1	0	1	2
	5	8	9	7	0
β	0	0	0	0	

Öka α_3 så mycket, vilket minskar rad 3.

Ungerska metoden: Exempel

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

					α
	1	2	0	3	2
	2	3	0	2	5
	0	1	0	1	2
	5	8	9	7	0
β	0	0	0	0	

Finna minsta element i rad 4.

Ungerska metoden: Exempel

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

					α
	1	2	0	3	2
	2	3	0	2	5
	0	1	0	1	2
	0	3	4	2	5
β	0	0	0	0	

Öka α_4 så mycket, vilket minskar rad 4.

Ungerska metoden: Exempel

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

					α
	1	2	0	3	2
	2	3	0	2	5
	0	1	0	1	2
	0	3	4	2	5
β	0	0	0	0	

Finn minsta element i kolumn 1. Blev noll.

Ungerska metoden: Exempel

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

					α
	1	2	0	3	2
	2	3	0	2	5
	0	1	0	1	2
	0	3	4	2	5
β	0	0	0	0	

Finn minsta element i kolumn 2.

Ungerska metoden: Exempel

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

					α
	1	1	0	3	2
	2	2	0	2	5
	0	0	0	1	2
	0	2	4	2	5
β	0	1	0	0	

Öka β_2 så mycket, vilket minskar kolumn 2.

Ungerska metoden: Exempel

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

					α
	1	1	0	3	2
	2	2	0	2	5
	0	0	0	1	2
	0	2	4	2	5
β	0	1	0	0	

Finn minsta element i kolumn 3. Blev noll.

Ungerska metoden: Exempel

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

					α
	1	1	0	3	2
	2	2	0	2	5
	0	0	0	1	2
	0	2	4	2	5
β	0	1	0	0	

Finn minsta element i kolumn 4.

Ungerska metoden: Exempel

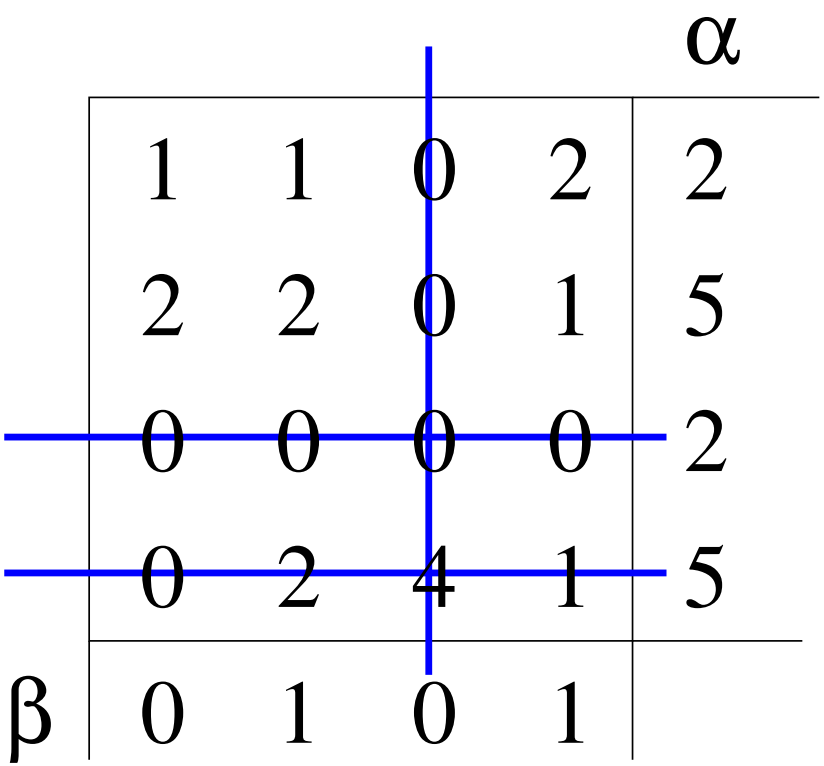
$$\hat{c}_{ij} = c_{ij} - \alpha_i - \beta_j$$

					α
	1	1	0	2	2
	2	2	0	1	5
	0	0	0	0	2
	0	2	4	1	5
β	0	1	0	1	

Öka β_4 så mycket, vilket minskar kolumn 4.

Ungerska metoden: Exempel

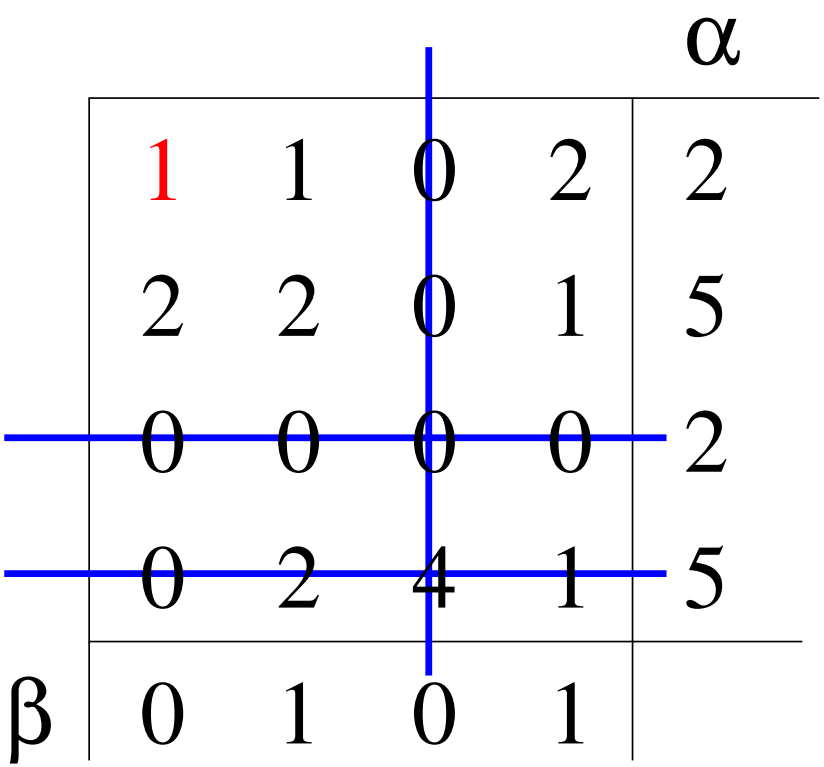
$$\hat{c}_{ij} = c_{ij} - \alpha_i - \beta_j$$



Stryk alla nollor. Stryk rad 3 och 4 samt kolumn 3. Tre streck.

Ungerska metoden: Exempel

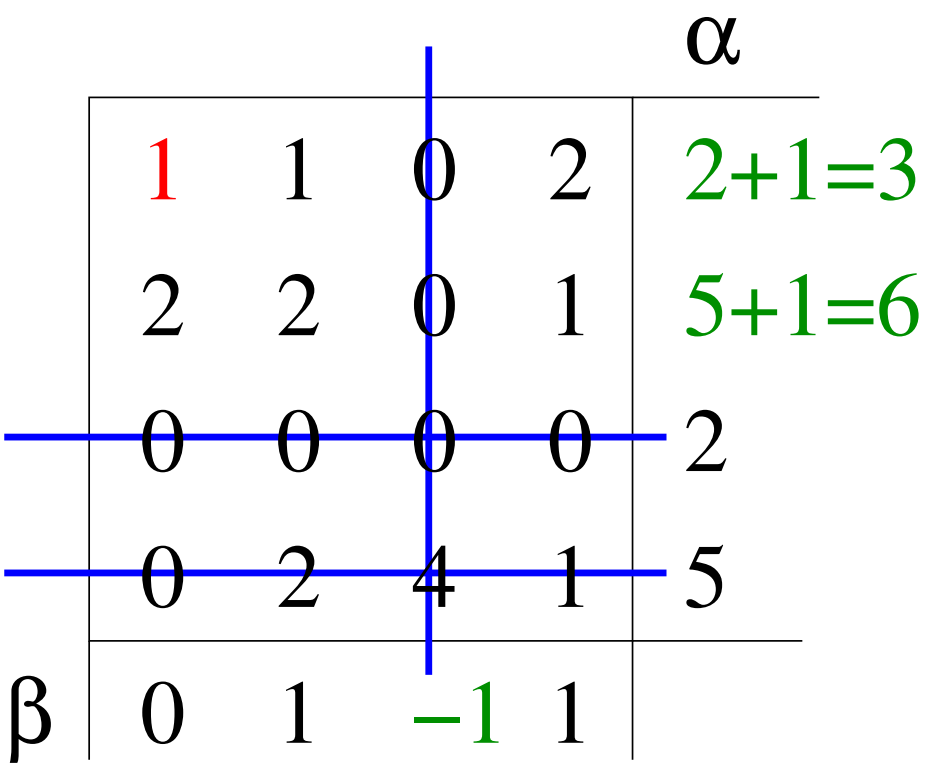
$$\hat{c}_{ij} = c_{ij} - \alpha_i - \beta_j$$



Minsta ostrukna element är 1.

Ungerska metoden: Exempel

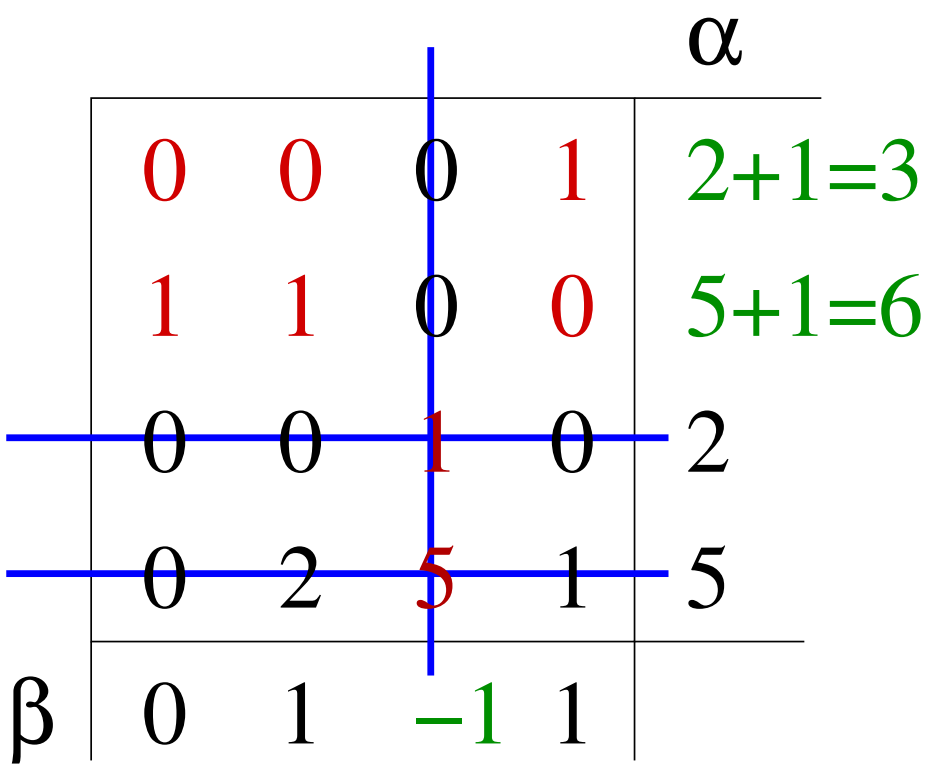
$$\hat{c}_{ij} = c_{ij} - \alpha_i - \beta_j$$



Öka α för ostrukna rader, minska β för strukna kolumner.

Ungerska metoden: Exempel

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



Alla ostrukna element minskas och dubbelt strukna ökas.

Ungerska metoden: Exempel

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

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

Nu går det inte att stryka alla nollor med färre än 4 streck.

Ungerska metoden: Exempel

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

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

Finn ensam nolla i rad eller kolumn.

Ungerska metoden: Exempel

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

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

Sätt $x = 1$ där. Inga fler nollor i denna rad och kolumn.

Ungerska metoden: Exempel

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

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

Finn ensam nolla i rad eller kolumn.

Ungerska metoden: Exempel

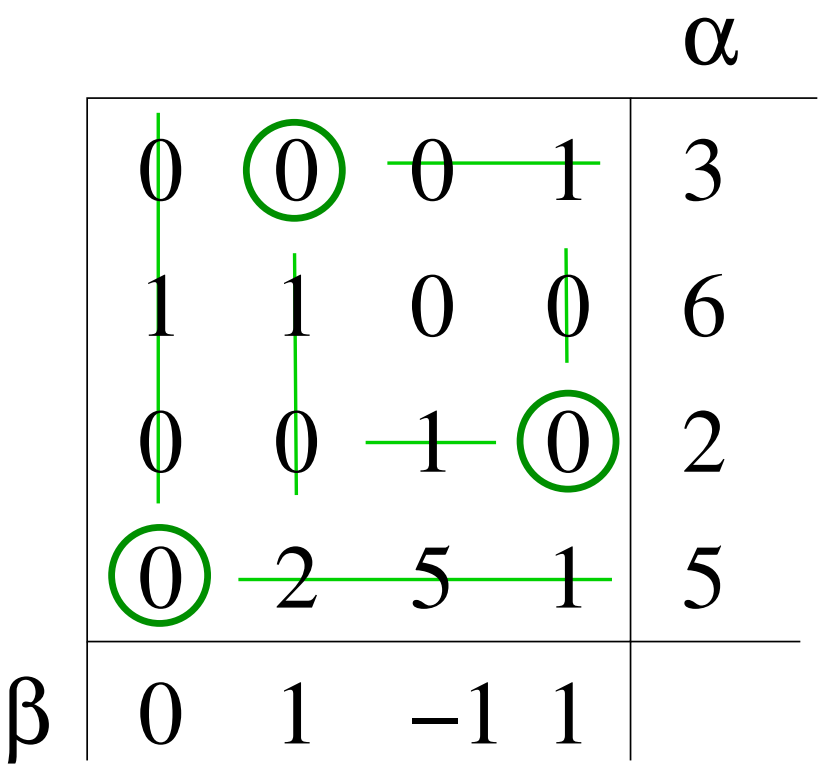
$$\hat{c}_{ij} = c_{ij} - \alpha_i - \beta_j$$

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

Sätt $x = 1$ där. Inga fler nollor i denna rad och kolumn.

Ungerska metoden: Exempel

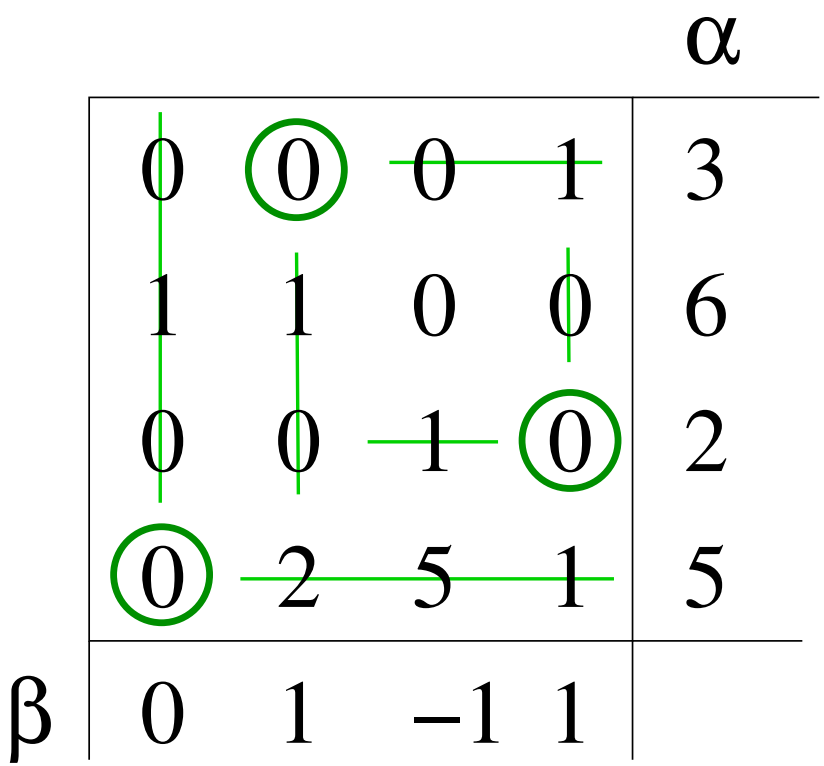
$$\hat{c}_{ij} = c_{ij} - \alpha_i - \beta_j$$



Finn ensam nolla i rad eller kolumn.

Ungerska metoden: Exempel

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



Sätt $x = 1$ där. Inga fler nollor i denna rad och kolumn.

Ungerska metoden: Exempel

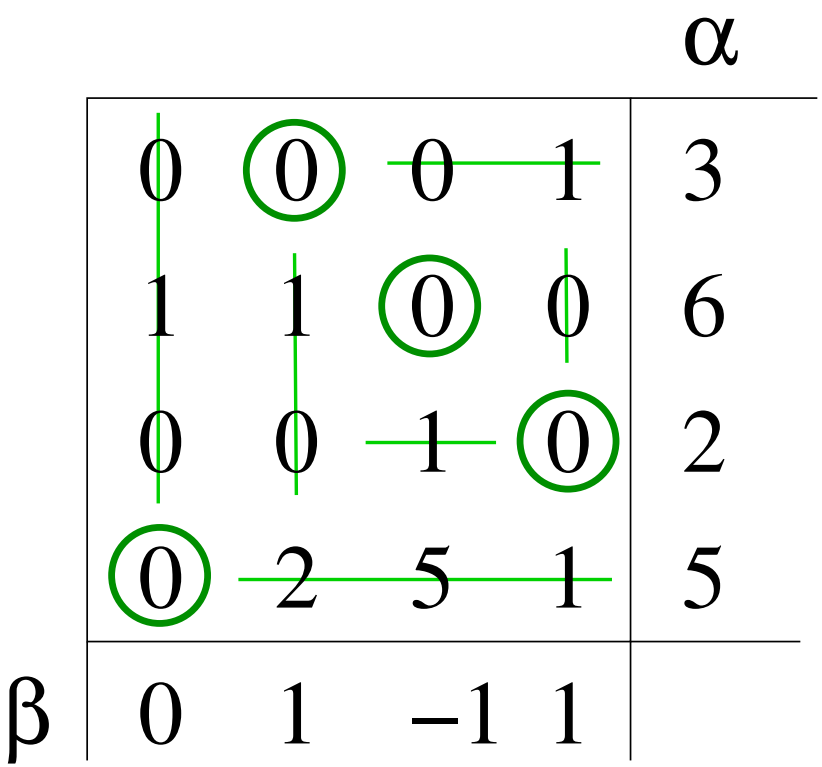
$$\hat{c}_{ij} = c_{ij} - \alpha_i - \beta_j$$

					α
	0	(0)	0	1	3
	1	1	(0)	0	6
	0	0	1	(0)	2
	(0)	2	5	1	5
β	0	1	−1	1	

Till slut återstår bara en nolla.

Ungerska metoden: Exempel

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



Sätt $x = 1$ där.

Ungerska metoden: Exempel

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

					α
	0	(0)	0	1	3
	1	1	(0)	0	6
	0	0	1	(0)	2
	(0)	2	5	1	5
β	0	1	−1	1	

Nu har vi positionerna.

Ungerska metoden: Exempel

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

0	(1)	0	0
0	0	(1)	0
0	0	0	(1)
(1)	0	0	0

Sätt $x = 1$ där.

Ungerska metoden: Exempel

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

0	1	0	0
0	0	1	0
0	0	0	1
1	0	0	0

Nu har vi x-lösningen.

Ungerska metoden: Exempel

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

Ungerska metoden: Exempel

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.

Ungerska metoden: Exempel

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

Ungerska metoden: Exempel

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:
$\alpha = (3, 6, 2, 5)$
$\beta = (0, 1, -1, 1)$

Ungerska metoden: Exempel

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

Ungerska metoden: Exempel

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.

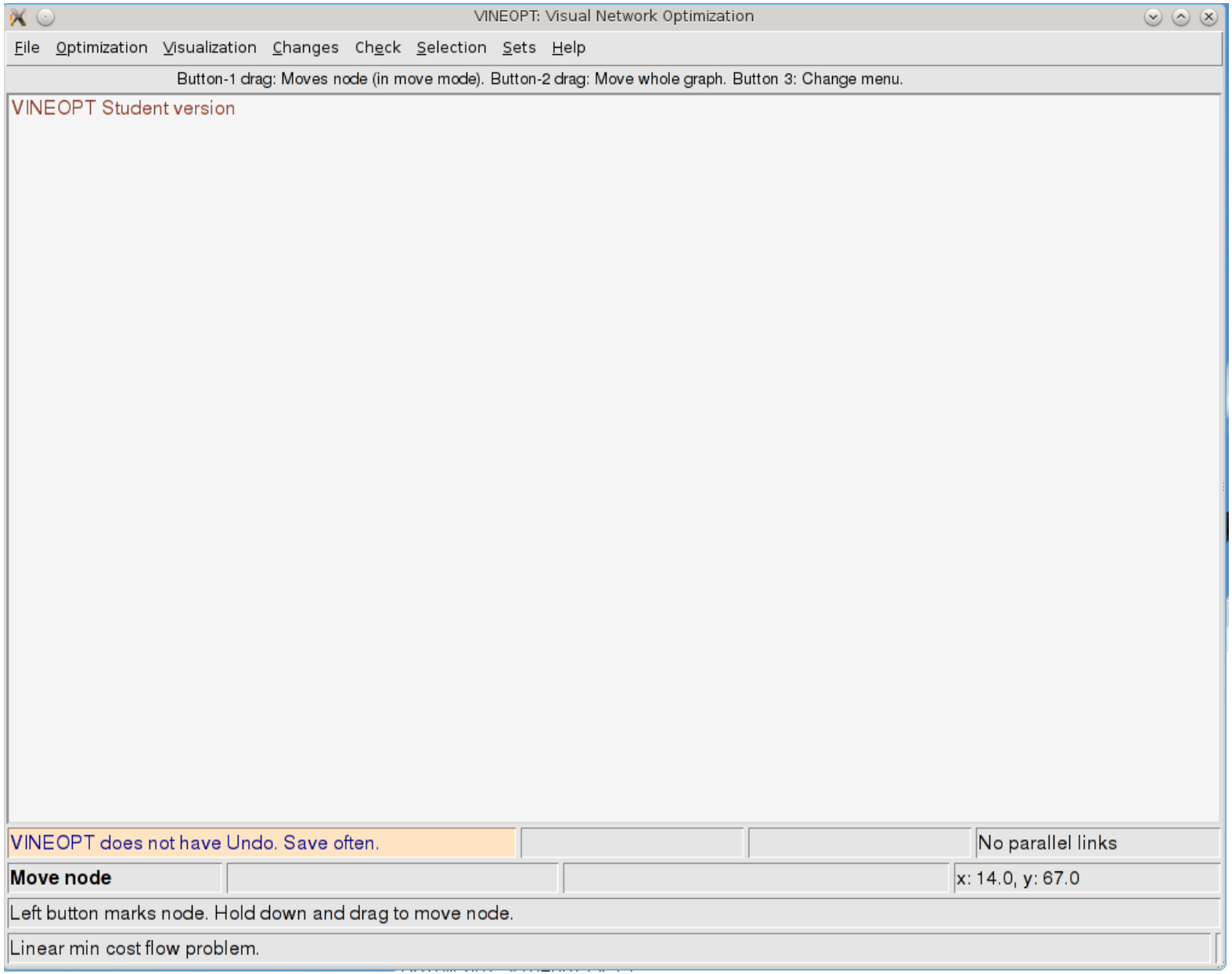
Total kostnad: 17

Duallösning:
$\alpha = (3, 6, 2, 5)$
$\beta = (0, 1, -1, 1)$

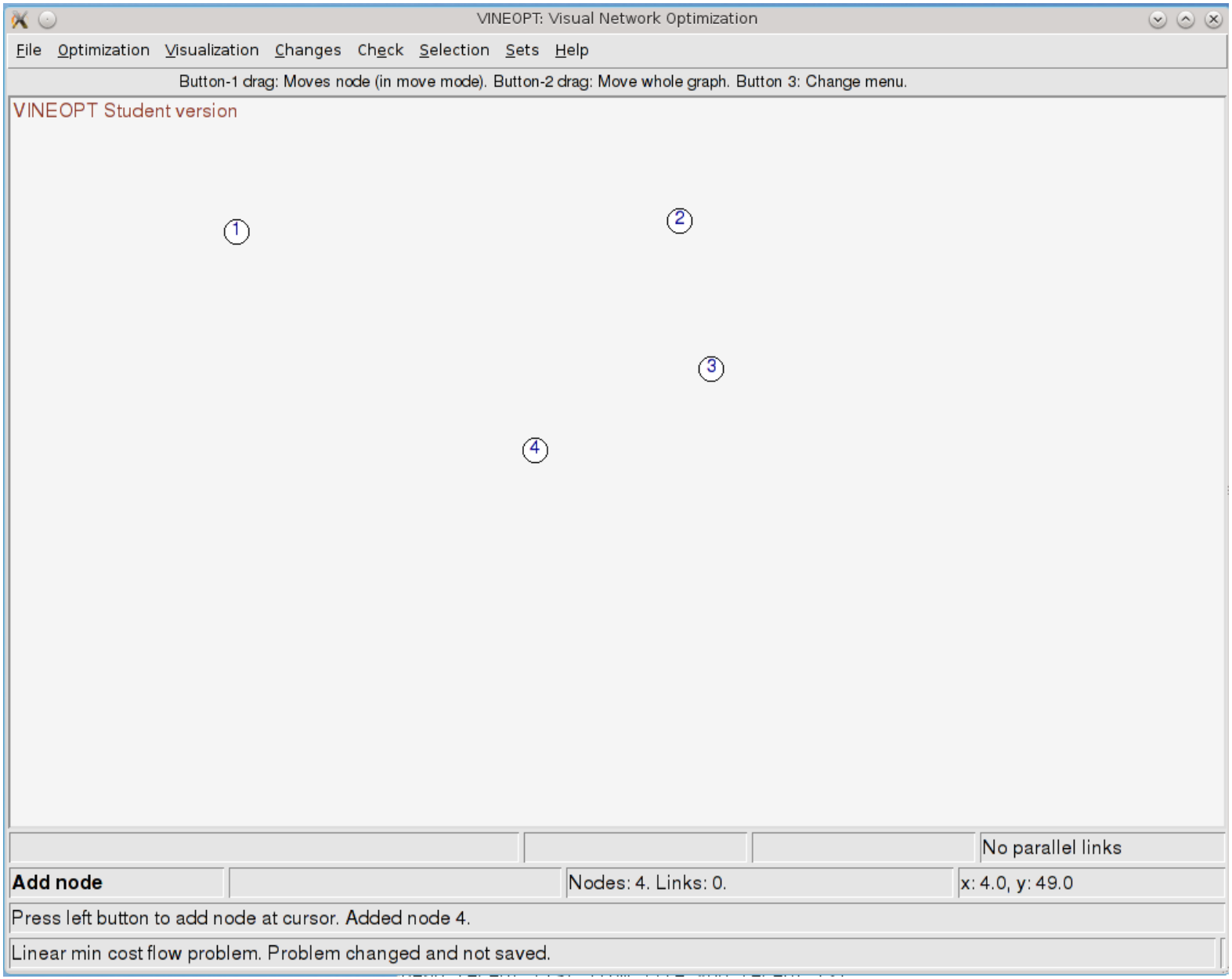
Kolla gärna starka dualsatsen: $v = \sum_{i=1}^{n} \alpha_i + \sum_{j=1}^{n} \beta_j$

Summan av dualvarialerna ska vara lika med totala kostnaden.

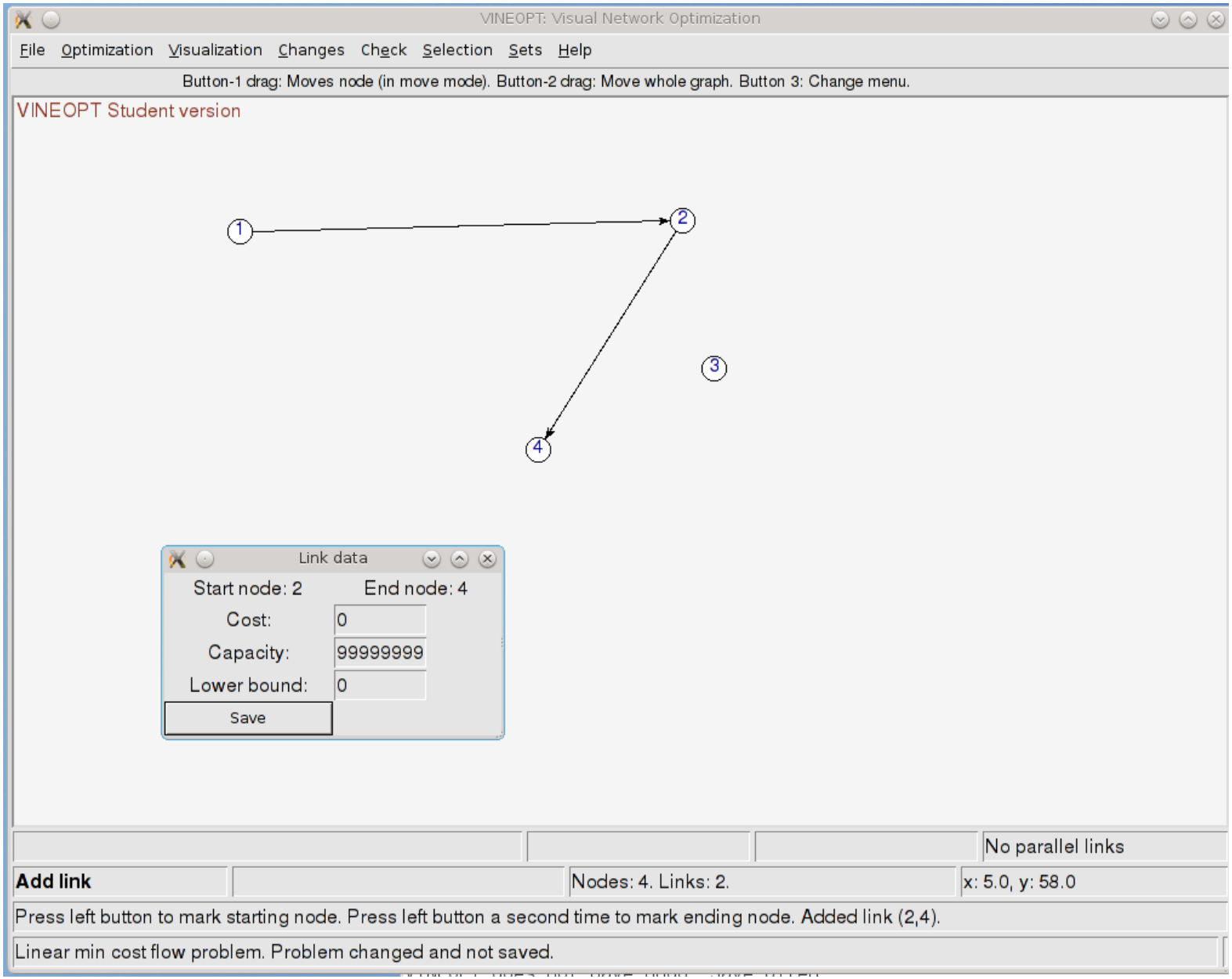
Vineopt


VINEOPT: Visual Network Optimization
File Optimization Visualization Changes Check Selection Sets Help
Button-1 drag: Moves node (in move mode). Button-2 drag: Move whole graph. Button 3: Change menu.
VINEOPT Student version
VINEOPT does not have Undo. Save often.
No parallel links
Move node
x: 14.0, y: 67.0
Left button marks node. Hold down and drag to move node.
Linear min cost flow problem.

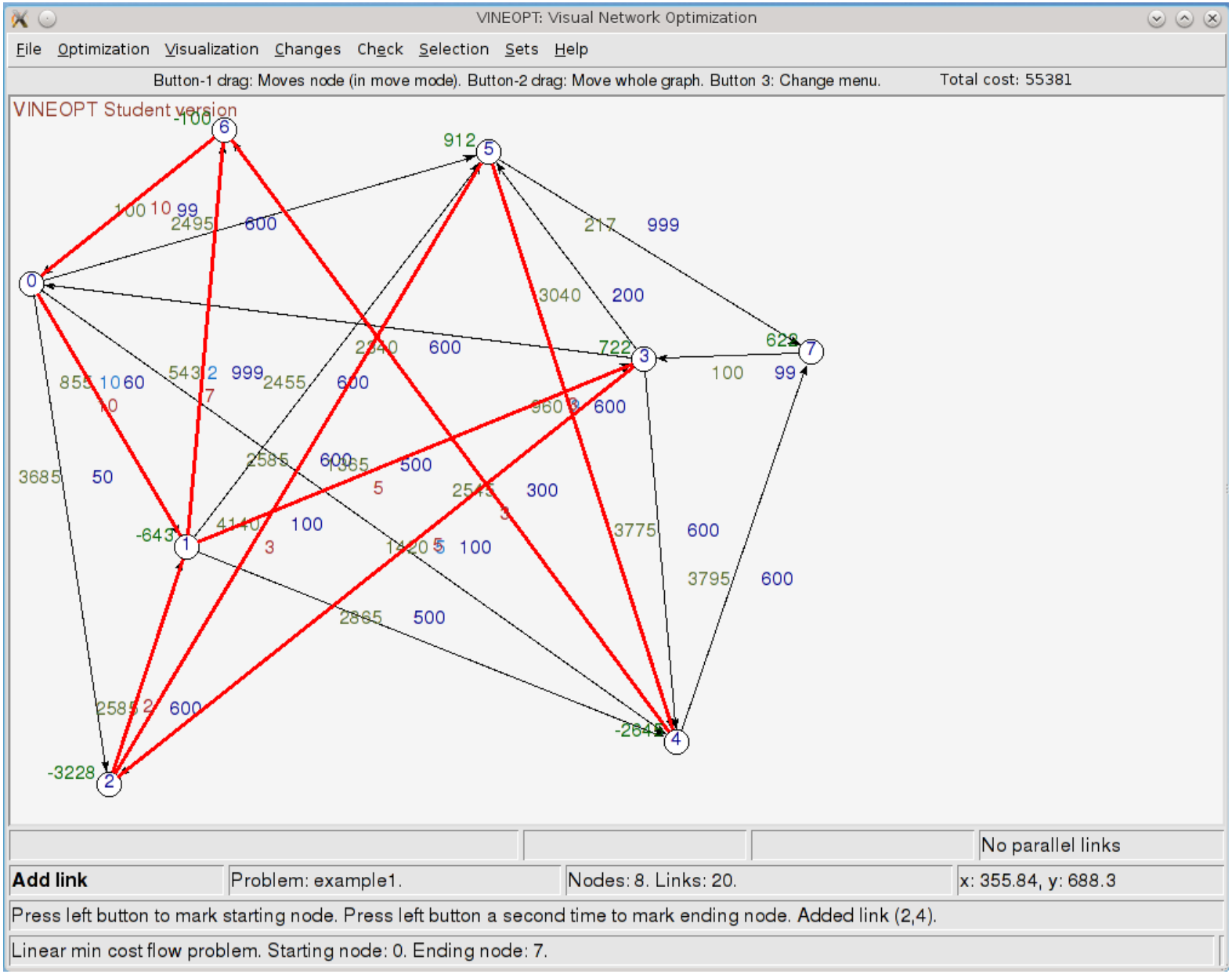
Vineopt


VINEOPT: Visual Network Optimization
File Optimization Visualization Changes Check Selection Sets Help
Button-1 drag: Moves node (in move mode). Button-2 drag: Move whole graph. Button 3: Change menu.
VINEOPT Student version
1
2
3
4
No parallel links
Add node
Nodes: 4. Links: 0.
x: 4.0, y: 49.0
Press left button to add node at cursor. Added node 4.
Linear min cost flow problem. Problem changed and not saved.

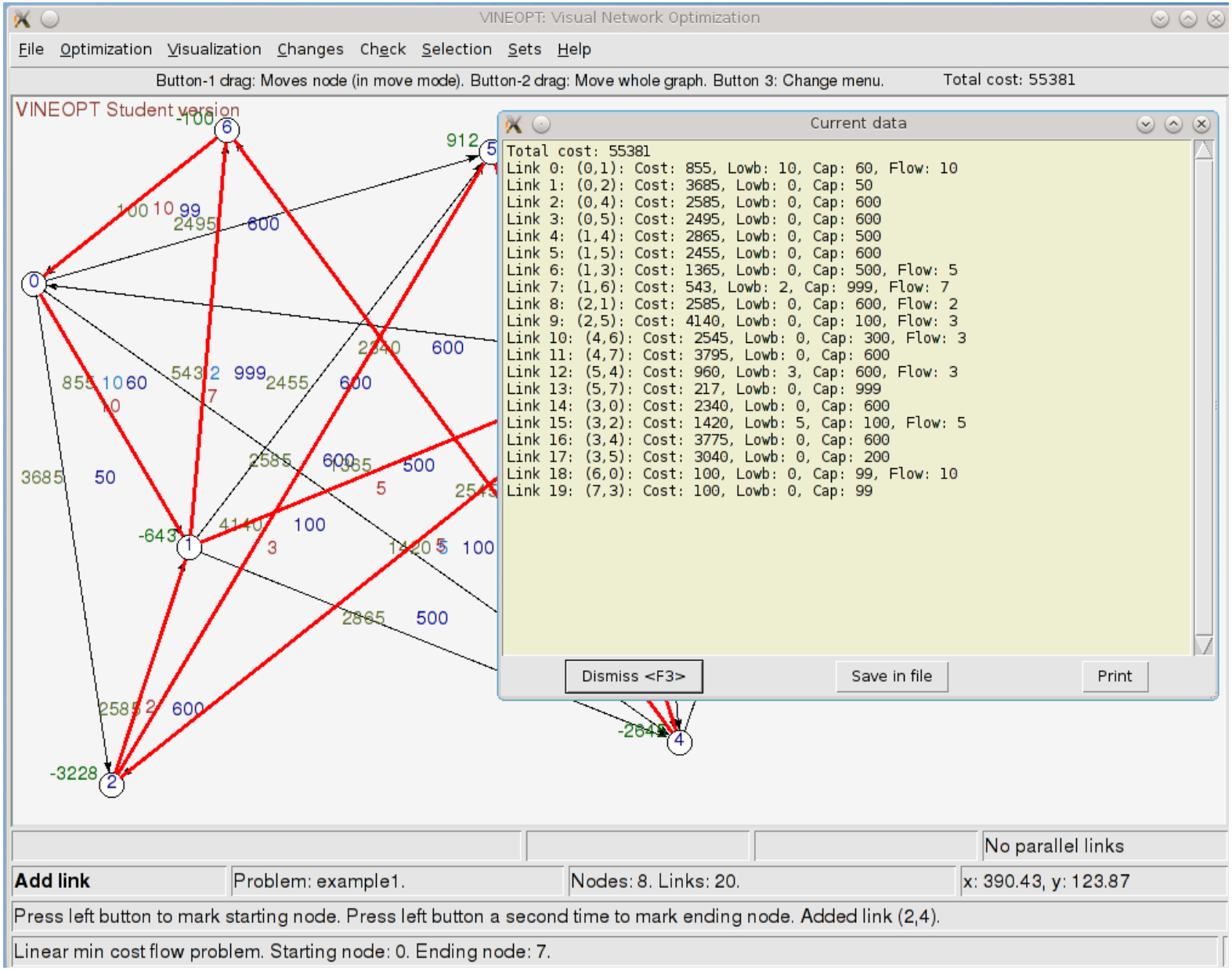
Vineopt


VINEOPT: Visual Network Optimization
File Optimization Visualization Changes Check Selection Sets Help
Button-1 drag: Moves node (in move mode). Button-2 drag: Move whole graph. Button 3: Change menu.
VINEOPT Student version
1
2
3
4
Link data
Start node: 2
End node: 4
Cost: 0
Capacity: 99999999
Lower bound: 0
Save
No parallel links
Add link
Nodes: 4. Links: 2.
x: 5.0, y: 58.0
Press left button to mark starting node. Press left button a second time to mark ending node. Added link (2,4).
Linear min cost flow problem. Problem changed and not saved.

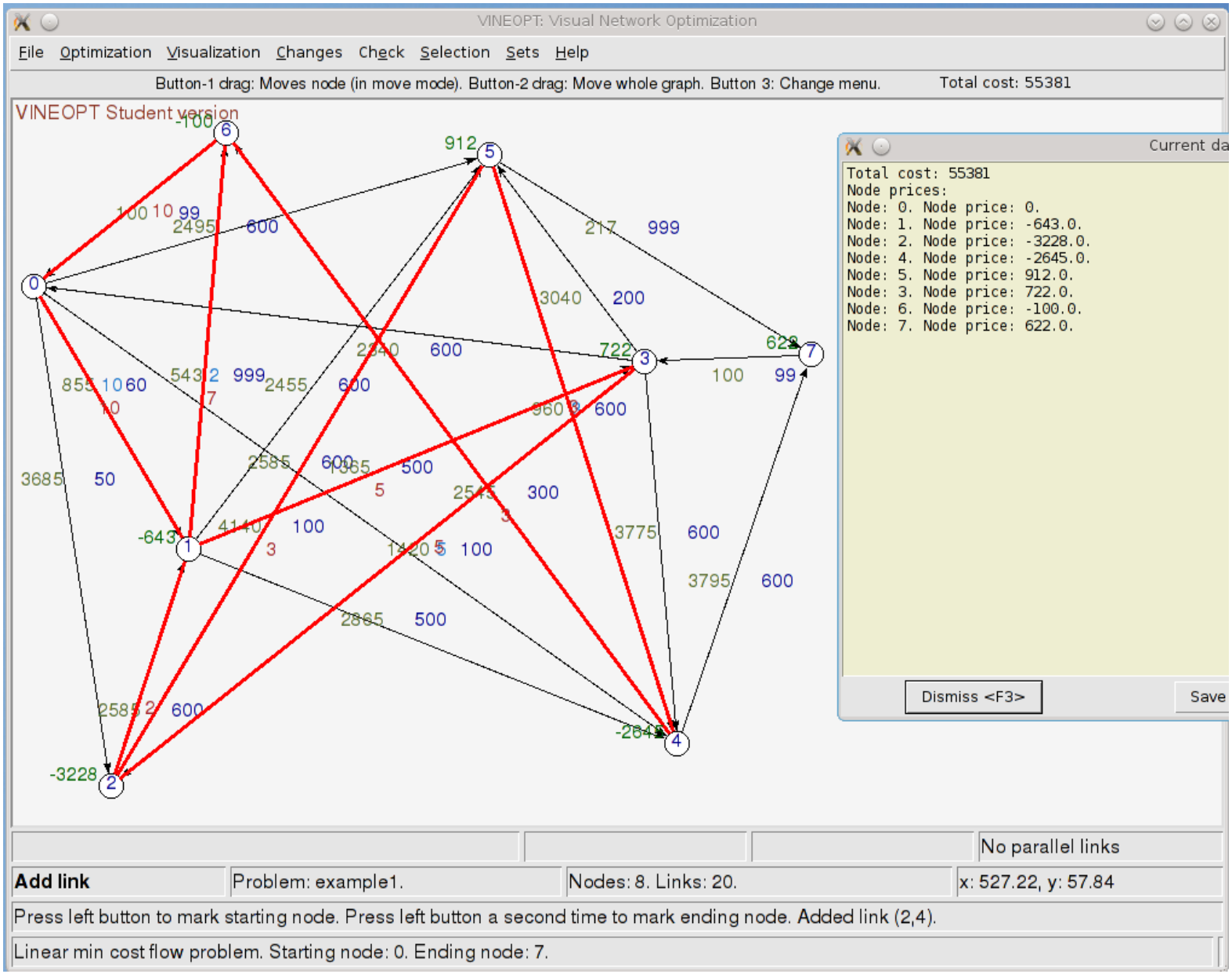
Vineopt


VINEOPT: Visual Network Optimization
File Optimization Visualization Changes Check Selection Sets Help
Button-1 drag: Moves node (in move mode). Button-2 drag: Move whole graph. Button 3: Change menu.
Total cost: 55381
VINEOPT Student version
No parallel links
Add link
Problem: example1.
Nodes: 8. Links: 20.
x: 355.84, y: 688.3
Press left button to mark starting node. Press left button a second time to mark ending node. Added link (2,4).
Linear min cost flow problem. Starting node: 0. Ending node: 7.

Vineopt


VINEOPT: Visual Network Optimization
File Optimization Visualization Changes Check Selection Sets Help
Button-1 drag: Moves node (in move mode). Button-2 drag: Move whole graph. Button 3: Change menu.
Total cost: 55381
VINEOPT Student version
Current data
Total cost: 55381
Link 0: (0,1): Cost: 855, Lowb: 10, Cap: 60, Flow: 10
Link 1: (0,2): Cost: 3685, Lowb: 0, Cap: 50
Link 2: (0,4): Cost: 2585, Lowb: 0, Cap: 600
Link 3: (0,5): Cost: 2495, Lowb: 0, Cap: 600
Link 4: (1,4): Cost: 2865, Lowb: 0, Cap: 500
Link 5: (1,5): Cost: 2455, Lowb: 0, Cap: 600
Link 6: (1,3): Cost: 1365, Lowb: 0, Cap: 500, Flow: 5
Link 7: (1,6): Cost: 543, Lowb: 2, Cap: 999, Flow: 7
Link 8: (2,1): Cost: 2585, Lowb: 0, Cap: 600, Flow: 2
Link 9: (2,5): Cost: 4140, Lowb: 0, Cap: 100, Flow: 3
Link 10: (4,6): Cost: 2545, Lowb: 0, Cap: 300, Flow: 3
Link 11: (4,7): Cost: 3795, Lowb: 0, Cap: 600
Link 12: (5,4): Cost: 960, Lowb: 3, Cap: 600, Flow: 3
Link 13: (5,7): Cost: 217, Lowb: 0, Cap: 999
Link 14: (3,0): Cost: 2340, Lowb: 0, Cap: 600
Link 15: (3,2): Cost: 1420, Lowb: 5, Cap: 100, Flow: 5
Link 16: (3,4): Cost: 3775, Lowb: 0, Cap: 600
Link 17: (3,5): Cost: 3040, Lowb: 0, Cap: 200
Link 18: (6,0): Cost: 100, Lowb: 0, Cap: 99, Flow: 10
Link 19: (7,3): Cost: 100, Lowb: 0, Cap: 99
Dismiss <F3>
Save in file
Print
No parallel links
Add link
Problem: example1.
Nodes: 8. Links: 20.
x: 390.43, y: 123.87
Press left button to mark starting node. Press left button a second time to mark ending node. Added link (2,4).
Linear min cost flow problem. Starting node: 0. Ending node: 7.

Vineopt


VINEOPT: Visual Network Optimization
File Optimization Visualization Changes Check Selection Sets Help
Button-1 drag: Moves node (in move mode). Button-2 drag: Move whole graph. Button 3: Change menu.
Total cost: 55381
VINEOPT Student version
Current da
Total cost: 55381
Node prices:
Node: 0. Node price: 0.
Node: 1. Node price: -643.0.
Node: 2. Node price: -3228.0.
Node: 4. Node price: -2645.0.
Node: 5. Node price: 912.0.
Node: 3. Node price: 722.0.
Node: 6. Node price: -100.0.
Node: 7. Node price: 622.0.
Dismiss <F3>
Save
No parallel links
Add link
Problem: example1.
Nodes: 8. Links: 20.
x: 527.22, y: 57.84
Press left button to mark starting node. Press left button a second time to mark ending node. Added link (2,4).
Linear min cost flow problem. Starting node: 0. Ending node: 7.