LP. Leksjon 8: Kapittel 13: Nettverk strøm problemer, forts.1

Vi fortsetter studiet av (MKS): minimum kost nettverk strøm problemet.

- Har nå en algoritme for beregning av x for gitt spenntre T
- Skal forklare hvorfor spenntrær svarer til LP basiser
- Skal også se hvordan de øvrige beregninger i en pivotering kan utføres

La A være node-kant indikator matrisen til (den rettede) grafen $D = (V, E)$. Vi har følgdende resultat:

Proposisjon

$$\text{rank }(A) = n - 1$$

der $n = |V|$ når D er sammenhengende (hvilket vi har antatt).

Bevis: Siden summen av alle radvektorene er nullvektoren, er radene i A lineært avhengige. Videre, hvis en av radene fjernes vil den resulterende $(n-1) \times m$ matrisen $\tilde{A}$ ha full radrang: grunnen er at submatrisen B av $\tilde{A}$ bestående av kolonnene som svarer til et spenntre vil være ikkesingulær. Faktisk vil en slik matrise B, etter passende permutasjoner av rader og kolonner, være triangulær og ha bare ± 1 på diagonalen. Og da er jo determinanten ± 1 så B er ikkesingulær. □

Oppgave:

1. Sjekk denne egenskapen ved B for en passende liten graf.
2. Bevis egenskapen generelt.

Betrakt igjen matrisen $\tilde{A}$ over, og la r være noden som svarer til den raden vi slettet i A; man kaller gjerne r rotnoden (for spenntrærne som kommer). La $\tilde{b}$ framkomme fra tilførsels/behovs-vektoren b ved at vi sletter komponenten som svarer til rotnoden r. De opprinnelige strømbalanse likningene $Ax = -b$ er ekvivalent med

$$\tilde{A}x = -\tilde{b}$$

fordi vi har slettet en redundant likning. (Den slettede raden i A er en lineær kombinasjon av radene i $\tilde{A}$ og fordi $\sum_v b_v = 0$ er b_r en tilsvarende lineær kombinasjon av de øvrige b_v-ene.)

Med denne omskrivningen er (MKS) problemet

$$\begin{array}{ll} \min & c^T x \\ \text{f.a.} & \\ & \tilde{A}x = -\tilde{b} \\ & x \geq O. \end{array}$$

og koeffisientmatrisen $\tilde{A}$ har full radrang. Nå er vi "in LP business"! La $N = n - 1$, så $\tilde{A}$ er en $N \times m$ matrise, og her er $m \geq N$ siden D er sammenhengende.

Minner om at en basis i $\tilde{A}$ er en ikkesingulær $N \times N$ submatrise av $\tilde{A}$; en slik matrise svarer altså til N utvalgte kolonner eller kanter i grafen og B er en basis nettopp når disse kantene danner et spenntre.

Theorem 13.1 En $N \times N$ kvadratisk submatrise B av $\tilde{A}$ er en basis hvis og bare hvis kolonnene i B svarer til et spenntre i D.

Bevis: Hvis kolonnene svarer til et spenntre, kan vi som nevnt permutere over i en triangulær matrise med ± 1 på diagonalen; dette skjer ved å ordne noder og kanter i henhold til stadig eliminasjon av et blad i treet. Hvis kolonnene ikke svarer til et spenntre, så må disse kantene inneholde en syklus (fordi det er $n-1$ kanter), og de tilhørende kolonnene i $\tilde{A}$ er lineært avhengige (summen av radvektorene er nullvektoren). $\square$

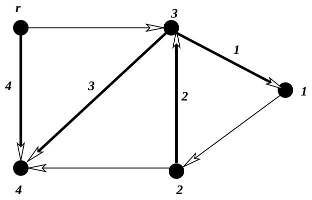
Så: det er en énentydig korrespondanse mellom spenntrær i grafen og LP basiser (i $\tilde{A}$).

Dessuten ser vi fra beviset:

- strukturen på hver basis B, triangulær med ± 1 på diagonalen innebærer at lineære likningssystemer med B eller B^T som koeffisientmatrise blir enkle å løse ved tilbakesubstitusjon og uten multiplikasjon/divisjon. Skal snart se detaljene i dette.

Dette er hovedgrunnen til at (MKS) problemer kan løses svært effektivt med simpleksalgoritmen.

Eksempel: Se på grafen under og spenntreet T (tykke linjer) der tallene angir nummerering av noder og kanter (r er rotnoden) ut fra bladeliminasjon.



Den tilhørende basisen B (med rader og kolonner nummerert som angitt) blir

$$B = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ -1 & 1 & -1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}.$$

Ser nå nærmere på simpleksalgoritmen for (MKS). Først partisjonerer vi $\tilde{A}$ ved

$$\tilde{A} = [B \;\; N]$$

der B er basisen som svarer til et spenntre T. (Egentlig er det en kolonnepermutasjon av $\tilde{A}$ som er lik matrisen på høyre side.) Så da er $\tilde{A}x = -\tilde{b}$ ekvivalent med

$$x_B = B^{-1}(-\tilde{b}) - B^{-1}Nx_N, \quad x_N \text{ fri}$$

der variabel vektoren x er partisjonert tilsvarende i basisdelen x_B og ikkebasisdelen x_N.

Den tilhørende basisløsningen er da gitt ved

$$x_B = B^{-1}(-\tilde{b}), \quad x_N = O.$$

Vi har allerede funnet algoritme som beregner x_B for et gitt spenntre T (og tilhørende basis B):

Algoritme for beregning av x_B for gitt T:

1. Velg et blad i spenntreet T dvs. en node som er inntil nøyaktig én kant e i T.
2. Beregn x_e for denne kanten e.
3. "Fjern" e fra T, og gå tilbake til steg 1 over inntil alle variablene er bestemt.

Men vi trenger også å beregne de duale variable knyttet til basisen B. *Hvordan gjør vi det?*

Husk at det duale problemet er

$$\begin{array}{ll} \max & -\sum_{v \in V} b_v y_v \\ \text{f.a.} & \end{array}$$

$$\begin{array}{rl} y_v - y_u + z_{uv} = c_{uv} & ((u,v) \in E) \\ z_{uv} \geq 0 & ((u,v) \in E). \end{array}$$

Vi finner nå de duale variablene slik:

- La $y_r = 0$. Egentlig har vi ingen dual variabel for rotnoden r siden den er slettet. Men siden vi bare har differenser av de duale variable i likningene over, vil også $y_v + \Delta$ oppfylle likningene hvis y_v gjør det. Dermed kan vi "normalisere" på en slik måte at dette stemmer med $y_r = 0$.

- Ved komplementær slakk er $z_{uv} = 0$ for hver $(u, v) \in E(T)$ (alternativt: hver slik $z_{uv} = 0$ er ikkebasisvariabel i det duale). Så fra likningene i det duale får vi $c_{uv} = y_v - y_u + z_{uv} = y_v - y_u$ dvs

$$y_v - y_u = c_{uv} \quad ((u, v) \in E(T))$$

Ved å starte i rotnoden r og arbeide oss gjennom T ved bladeliminasjon kan vi bruke likningnene til å beregne y_v-ene én etter én.

3. Den primale nettverk simpleks algoritmen

Vi starter med et spenntre T der den tilhørende treløsningen x_B er tillatt, dvs. $x_B \geq O$. Vi forklarer hvordan man finner et slikt tre senere.

Algoritmen kan oppsummeres slik:

- Sjekk optimalitet ved å beregne duale variable y og z og sjekke om $z \geq O$.
- Hvis ikke optimal: gjennomfør en pivotering. Dette skjer ved å velge en kant $e = (u, v)$ der $z_{uv} < 0$ (dvs. negativ redusert kost), og finne et nytt spenntre T' ved å legge til e og fjerne en viss annen kant (slik at T' blir et spenntre). Oppdater treløsningen x.

Kommentarer:

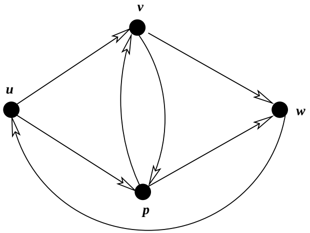
- Beregningen av både x og y skjer ved "bladeliminasjon" ved å bruke triangulariteten slik vi har sett. Beregningen av hver z_{uv} skjer direkte fra den tilhørende likningen i det duale problemet:

$$z_{uv} = c_{uv} + y_u - y_v.$$

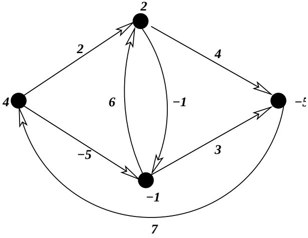
- Så for hver kant (u, v) utenfor treet gjør vi denne (enkle) beregningen. (Likevel er det dette som tar mest tid for virkelig store problemer i grafer med mange kanter.)

Vi ser nærmere på metoden for et lite eksempel.

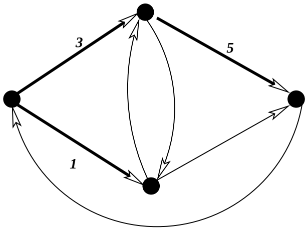
Eksempel. Følgende figur viser grafen. La w være rotnode.



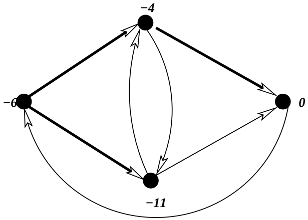
og her er verdiene på b og c



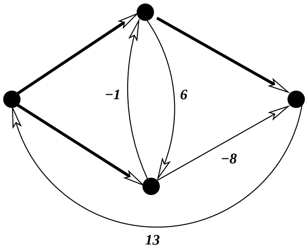
1. iterasjon: Vi velger et spenntre (tykke linjer i neste figur) og beregner x (også angitt) som blir tillatt.



Deretter beregnes y:



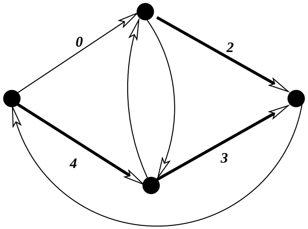
Ut fra y og opprinnelige kostnader c finner vi z (angir disse bare for kanter utenfor T; i treet er jo $z_{uv} = 0$):



Løsningen er ikke optimal, f.eks. er $z_{pw} = -8 < 0$. Vi velger nå å ta denne variabelen inn i basis. Så denne kanten skal inn i treet.

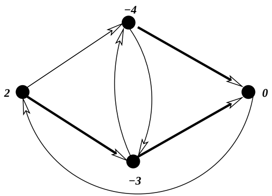
Hvor mye kan vi øke x_{pw}? Hvis vi lar $x_{pw} = \epsilon$, så finner vi at basisvariablene får verdiene $x_{vw} = 5 - \epsilon$, $x_{uv} = 3 - \epsilon$, $x_{up} = 1 + \epsilon$. Her har vi brukt likningene som uttrykker basisvariablene som funksjon av ikkebasisvariablene.

Vi ser at maksimal verdi på ϵ er 3; fordi en større verdi vil gi en ikketillatt primal basisløsning. For $\epsilon = 3$ vil de nye verdiene på basisvariablene være $x_{vw} = 2$, $x_{uv} = 0$, $x_{up} = 4$. Som ventet ble en av basisvariablene 0, nemlig x_{uv}, så vi oppdaterer basis ved å erstatte x_{uv} med x_{pw} i basis. Dette gir det nye spenntreet og tilhørende treløsning som angitt i neste figur:

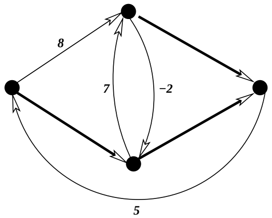


Vi har fullført iterasjon 1.

2. iterasjon: Beregner y:

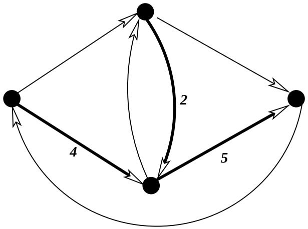


Beregner z:

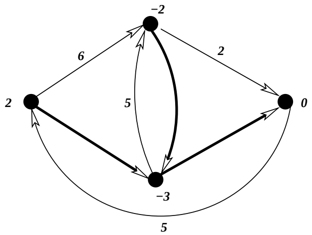


Løsningen er **ikke optimal** fordi $z_{vp} < 0$.

Vi føyer nå kanten (v, p) til treet og får en syklus. Sender en strøm på ϵ i denne syklusen i retning som (v, p). Resultatet er at maksimal ϵ er 2 og da blir strømmen i (v, w) lik 0. Så treet oppdateres ved at (v, w) erstattes av (v, p). Den oppdaterte treløsningen x er da:



3. iterasjon: Så beregner vi igjen y og z som blir (vist i samme figur):



Nå er $z \geq O$, så løsningen er optimal (både primal og dual løsning).
Problemet er løst!

I neste leksjon vil vi oppsummere metoden og komme med noen kommentarer.