

Masteroppgave

# Strøm i nettverk og transportmatriser med bestemte strukturer

**Ida Amelie Eid Robsahm**

Matematikk, Lektorprogrammet  
30 studiepoeng

Matematisk institutt  
Det matematisk-naturvitenskapelige fakultet

Våren 2023





**Ida Amelie Eid Robsahm**

Strøm i nettverk og transportmatriser  
med bestemte strukturer

Veileder:  
Geir Dahl



## Sammendrag

I denne oppgaven utforsker vi sammenhengen mellom strøm i nettverk og transportmatriser der matriseelementene oppfyller gitte begrensninger. Videre bruker vi denne sammenhengen til å avgjøre om det eksisterer, og i så fall bestemme, nedre triangulære, øvre triangulære og tridiagonale transportmatriser, samt transportmatriser der elementene enten er 0 eller 1. Deretter karakteriserer vi når disse typene matriser eksisterer uten å løse ett nettverk strøm problem. I forkant av den utforskende delen der dette diskuteres, innledes oppgaven med en teoridel. Her redegjør vi for relevant teori om strøm i nettverk, og beskriver ulike nettverk strøm problemer. Videre presenteres noen anvendelser av de introduserte nettverk strøm problemene. Teoridelen avsluttes med to kapitler der vi introduserer teori som karakteriserer når det eksisterer trappematiser og  $(0,1)$ -matriser med gitte linjesummer. I den utforskende delen av oppgaven kobles egne resultater til denne teorien.



# Innhold

Sammendrag	i
Oversikt over figurer	v
Anerkjennelser	vii
1 Innledning	1
1.1 Disposisjon	1
1.2 Egne bidrag	1
<b>I Bakgrunnsteori og Anvendelser</b>	<b>3</b>
2 Grafteori	5
2.1 Terminologi	5
2.2 Matching	6
3 Nettverk strøm teori	7
3.1 Strøm og sirkulasjon	7
3.2 Maksimum strøm problemet	9
3.3 Minimum kutt problemet	12
3.4 Det tillatte strøm problemet	15
3.5 Minimum kost strøm problemet	17
4 Anvendelser av nettverk strøm teori	21
4.1 Matching med maksimum kardinalitet	21
4.2 Transportmatriser	23
4.3 Matriseavrundingsproblemet	27
5 Transportmatriser med trappemønster	33
5.1 Trappematriser	33
5.2 Strøm i nettverk og transportmatriser med trappemønster	34
6 (0,1)-matriser med gitte linjesummer	37
6.1 Strøm i nettverk og (0,1)-matriser med gitte linjesummer	37
6.2 Gale-Ryser teorem og algoritme	38
<b>II Presentasjon og diskusjon av egne resultater</b>	<b>41</b>
7 Eksistens av transportmatriser med bestemte strukturer	43
7.1 Strøm i nettverk og transportmatriser med bestemte strukturer	43

7.2	Eksistens av nedre triangulære transportmatriser	46
7.3	Eksistens av øvre triangulære transportmatriser	48
7.4	Eksistens av tridiagonale transportmatriser	49
7.5	MATLAB-program for å konstruere nedre triangulære, øvre triangulære og tridiagonale transportmatriser	51
8	Eksistens av $(0,1)$ -matriser med to, tre og fire rader og gitte linjesummer	53
8.1	Eksistens av $(0,1)$ -matriser med to rader og gitte linjesummer	53
8.2	Eksistens av $(0,1)$ -matriser med tre rader og gitte linjesummer	54
8.3	Eksistens av $(0,1)$ -matriser med fire rader og gitte linjesummer	56
8.4	Diskusjon av egne resultater mot Gale-Ryser teorem	60
8.5	MATLAB-program for å lage $(0,1)$ -matriser med to, tre og fire rader og gitte linjesummer	63
9	Konklusjon	65
	Referanser	67
A	Bevis tilhørende kapittel 7	69
A.1	Nordvest-Hjørne regelen produserer nedre triangulære transportmatriser	69
A.2	Nordvest-Hjørne regelen produserer øvre triangulære transportmatriser	72
A.3	Nordvest-Hjørne regelen produserer tridiagonale transportmatriser	75
B	MATLAB-kode tilhørende kapittel 7.5	79
B.1	Avgjør om det finnes en nedre triangulær transportmatrise	79
B.2	Avgjør om det finnes en øvre triangulær transportmatrise	79
B.3	Avgjør om det finnes en tridiagonal transportmatrise	80
B.4	Lager nedre triangulære, øvre triangulære og tridiagonale transportmatriser	81
C	MATLAB-kode tilhørende kapittel 8.5	85
C.1	Lager $(0,1)$ -matriser med to rader og gitte linjesummer	85
C.2	Lager $(0,1)$ -matriser med tre rader og gitte linjesummer	86
C.3	Lager $(0,1)$ -matriser med fire rader og gitte linjesummer	88
C.4	Returnerer en $(0,1)$ -matrise med to, tre eller fire rader og gitte linjesummer	92



## Figurer

2.1	En urettet og en rettet graf	5
2.2	En urettet og en rettet komplett bipartitt graf	6
2.3	En perfekt, maksimum og maksimal matching	6
3.1	Et nettverk med tall assosiert med både kanter og noder.	7
3.2	Et nettverk og tilhørende residualnettverk	10
3.3	Ford-Fulkerson algoritmen	11
3.4	To $st$ -kutt med ulike kapasiteter	12
3.5	Et minimum $st$ -kutt	14
3.6	Et nettverk og tilhørende transformert nettverk	15
3.7	Nettverk tilknyttet et tillatt strøm problem	16
3.8	Et nettverk tilknyttet et minimum kost strøm problem	17
3.9	Syklus-kansellering algoritmen	19
4.1	Et matchingproblem representert som en urettet bipartitt graf	21
4.2	En urettet bipartitt graf og korresponderende nettverk	22
4.3	En maksimum $st$ -strøm og korresponderende matching	23
4.4	Det korresponderende nettverket til en $3 \times 3$ transportmatrise	25
4.5	Det korresponderende nettverket til en $4 \times 5$ transportmatrise	26
4.6	En maksimum $st$ -strøm og korresponderende matrise	27
4.7	Matriseavrundingsproblem: korrespondansen mellom matrise og nettverk	28
4.8	Matriseavrundingsproblem: nettverk etter variabelskifte ( $x \rightarrow \tilde{x}$ )	29
4.9	Matriseavrundingsproblem: nettverk utvidet med noder $s'$ og $t'$	30
4.10	Matriseavrundingsproblem: residualnettverk til en maksimum $s't'$ -strøm $\tilde{x}$	30
4.11	Matriseavrundingsproblem: en tillatt strøm $\tilde{x}$	31
4.12	Matriseavrundingsproblem: en tillatt strøm $x$ og korresponderende matrise	31
5.1	Strøm i nettverk og en transportmatrise med gitt trappemønster	35
6.1	Strøm i nettverk og en $(0,1)$ -matrise med gitte linjesummer	38
7.1	Det transformerte nettverket til en $3 \times 3$ transportmatrise	44
7.2	En maksimum $st$ -strøm i nettverket til en diagonal transportmatrise	44
7.3	En maksimum $st$ -strøm i nettverket til en nedre triangulær transportmatrise	45
7.4	En maksimum $st$ -strøm i nettverket til en øvre triangulær transportmatrise	45
7.5	En maksimum $st$ -strøm i nettverket til en tridiagonal transportmatrise	46
7.6	Sammenhenger mellom linjesummer i nedre triangulære transportmatriser	47
7.7	Sammenhenger mellom linjesummer i tridiagonale transportmatriser	49
7.8	MATLAB-plot som korresponderer med en tridiagonal $5 \times 5$ transportmatrise	52



# Anerkjennelser

Jeg vil først rette en stor takk til min veileder Geir Dahl, som har inkludert oss lektorstudenter fra dag en. Takk for ukentlige mandagsmøter og for at døren din alltid har vært åpen for spørsmål. Jeg har lært mye av din oppmuntring til å utforske - en tilnærming til matematikken jeg er inspirert til å ta med meg inn i klasserommet. Videre må jeg takke alle professorer og gruppelærere på matematisk institutt som jeg har rent ned dørene til i løpet av de siste årene. Takk for at dere har tatt dere tid til å svare på spørsmål når jeg har stått fast. Det har motivert meg til å ville forstå, en holdning jeg håper smitter over på fremtidige elever.

Jeg vil også takke mamma og pappa som har vært mine største støtte- og lagspillere hele livet. Takk til storebroren min Marius for oppmuntring alle ganger jeg har ringt og krisemaksimert. Takk til Nora, Stine, Jenny og alle jeg har delt leseplass med de siste to årene. Lunsjer, utblåsninger og latter har gjort masterhverdagen bedre. Ikke minst må jeg takke terrengløpelaget jeg har trent med det siste året, Untold\_Movement. Intervaller med en rå gjeng hver tirsdag har vært et ukentlig høydepunkt. Til slutt må jeg takke gode venner utenfor lesesalen, for støtte og tålmodighet under hele studieløpet. Jeg gleder meg til mer tid sammen med dere fremover!



# Kapittel 1

## Innledning

### 1.1 Disposisjon

I første del av oppgaven redegjøres det for relevant teori, samt anvendelser av teorien. Videre følger en utforskende del som baserer seg på den introduserte teorien. Som en del av det utforskende arbeidet har jeg benyttet [MAT22]. Det har resultert i to MATLAB-koder, oppsummert i slutten av Kapittel 7 og Kapittel 8, og gjengitt i Tillegg B og Tillegg C.

### 1.2 Egne bidrag

I første del av oppgaven innebærer egne bidrag eksempler som illustrerer algoritmer og metoder for å løse ulike nettverk strøm problemer, samt eksempler som anvender introduserte teoremer. I forkant av disse eksemplene er det tydelig referert til teorien og metodene eksemplene er basert på. I andre del av oppgaven innebærer egne bidrag først et eksempel som illustrerer hvordan vi kan løse et nettverk strøm problem for å avgjøre om det eksisterer, og i så fall finne en diagonal, øvre triangulær, nedre triangulær og/eller tridiagonal transportmatrise. Videre innebærer egne bidrag følgende resultater:

**Teorem 7.2.2:** Karakteriserer når det eksisterer en ikke-negativ nedre triangulær matrise med gitte linjesummer.

**Teorem 7.3.1:** Karakteriserer når det eksisterer en ikke-negativ øvre triangulær matrise med gitte linjesummer.

**Teorem 7.4.2:** Karakteriserer når det eksisterer en ikke-negativ tridiagonal matrise med gitte linjesummer.

**Teorem 8.1.1:** Karakteriserer når det eksisterer en  $(0,1)$ -matrise med to rader og gitte linjesummer.

**Teorem 8.2.1:** Karakteriserer når det eksisterer en  $(0,1)$ -matrise med tre rader og gitte linjesummer.

**Teorem 8.3.1:** Karakteriserer når det eksisterer en  $(0,1)$ -matrise med fire rader og gitte linjesummer.

**Proposisjon 8.4.1:** Bekrefter at Teorem 8.1.1 kan betraktes som en forenkling av Gale-Ryser teorem for  $(0,1)$ -matriser med to rader.

**Proposisjon 8.4.2:** Bekrefter at Teorem 8.2.1 kan betraktes som en forenkling av Gale-Ryser teorem for  $(0,1)$ -matriser med tre rader.

**Proposisjon 8.4.3:** Bekrefter at Teorem 8.3.1 kan betraktes som en forenkling av Gale-Ryser teorem for  $(0,1)$ -matriser med fire rader.

Med unntak av deler av bevisene av Teorem 7.2.2, Teorem 7.3.1 og Teorem 7.4.2 som er gitt i Tillegg A, er alle teoremene/proposisjonene bevist i forkant eller etterkant av der de presenteres.

**Del I**

# Bakgrunnsteori og Anvendelser





## Kapittel 2

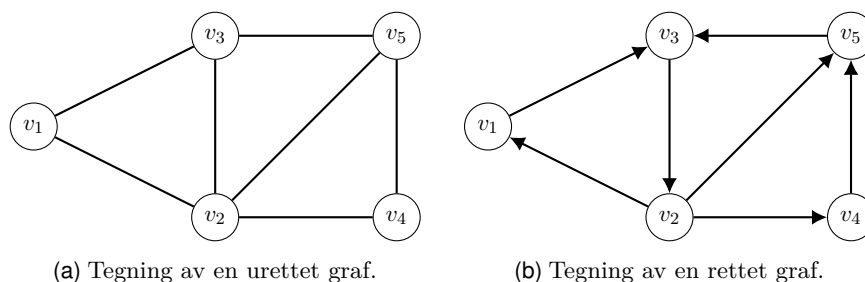
# Grafteori

### 2.1 Terminologi

I grafteori skilles det mellom urettede og rettede grafer, der den eneste forskjellen er hvorvidt kantene i grafen har retning eller ikke. Begrepet *graf* vil i denne oppgaven betegne en rettet graf, dersom ikke annet presiseres. Videre definisjoner og notasjon er hentet fra [BM08], hvis ikke annet oppgis.

En *graf*  $G$  er et ordnet par  $(V, E)$  bestående av en mengde noder  $V$ , og en mengde kanter  $E$ , sammen med en forbindelsesfunksjon (engelsk: incident function)  $\Psi_G$  som tilordner hver kant i  $E$  et par av noder i  $V$ . I denne oppgaven er det utelukkende *enkle grafer* som skal diskuteres. Det vil si grafer der ingen kant har begge sine endepunkter i samme node og der ingen to kanter forbinder samme to noder. For enkle grafer trenger vi ikke en forbindelsesfunksjon for å navngi kantene. Vi betegner heller en kant mellom to noder  $v_1$  og  $v_2$  med  $v_1v_2$  i en urettet og med  $(v_1, v_2)$  i en rettet graf.

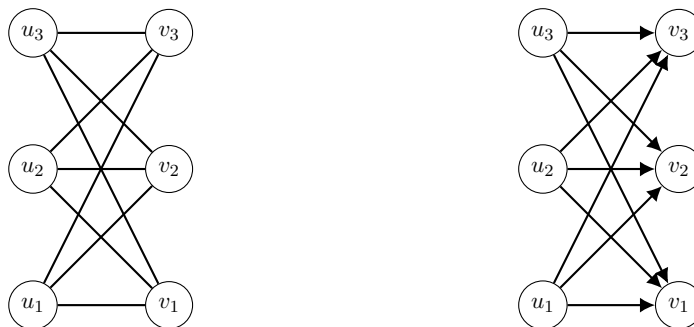
Hver graf kan representeres med en tegning i planet, der nodene betegnes med punkter og kantene med linjestykker mellom punktene. I en rettet graf bruker vi piler for å indikere retning. Figur 2.1a viser en tegning av en enkel urettet graf, mens Figur 2.1b viser en tegning av en enkel rettet graf, begge med samme node- og kantmengde.



Figur 2.1

En graf  $G' = (V', E')$  er en *delgraf* av  $G = (V, E)$  hvis  $V' \subseteq V$  og  $E' \subseteq E$ . En *vei* (engelsk: path) er en delgraf bestående av en følge av forskjellige noder,  $v_1 - v_2 - \dots - v_k$ , som er slik at etterfølgende noder i følgen er forbundet med en felles kant. I en *rettet vei*,  $v_1 - v_2 - \dots - v_k$ , er  $(v_{i-1}, v_i), (v_i, v_{i+1}) \in E$  for alle  $i = 2, \dots, k - 1$  [AMO93, s. 26].

En type graf som er relevant for senere diskusjon, er det som kalles en *bipartitt graf*. Det er en graf som er slik at nodemengden kan deles inn i to disjunkte mengder  $U$  og  $V$ , og der hver kant har en node i  $U$  og en node i  $V$ . Hvis hver node i  $U$  og hver node i  $V$  er forbundet, kalles  $G$  en komplett bipartitt graf. Figur 2.2 viser et eksempel på en urettet og en rettet komplett bipartitt graf.



(a) En komplett urettet bipartitt graf. (b) En komplett rettet bipartitt graf.

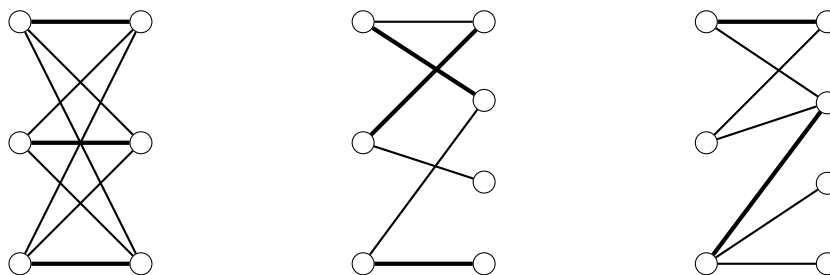
Figur 2.2

## 2.2 Matching

En sentral definisjon, mye anvendt i kombinasjon med urettede bipartitte grafer, er det som kalles *matching*. En matching i en graf  $G = (V, E)$  er en mengde kanter i  $E$  som er slik at hver node i  $V$ , maksimalt er inntil en av kantene i mengden [AMO93, s. 9]. For å senere diskutere matchingproblemer på en presis måte trenger vi følgende terminologi, hentet fra [BM08, s. 414]:

- En node som er inntil en kant i en matching, er dekket av matchingen.
- En perfekt matching er en som dekker alle nodene i grafen.
- En maksimum matching er en som dekker så mange noder som mulig i grafen.
- En matching er maksimal hvis den ikke kan utvides til en større matching.

Videre sier vi at en matching bestående av  $k$  kanter har kardinalitet lik  $k$  [AMO93]. De uthevede kantene i Figur 2.3a-c viser eksempler på en perfekt, maksimum og maksimal matching, definert på urettede bipartitte grafer. Det er enkelt å se at en perfekt matching også er maksimum, og at en maksimum matching også er maksimal.



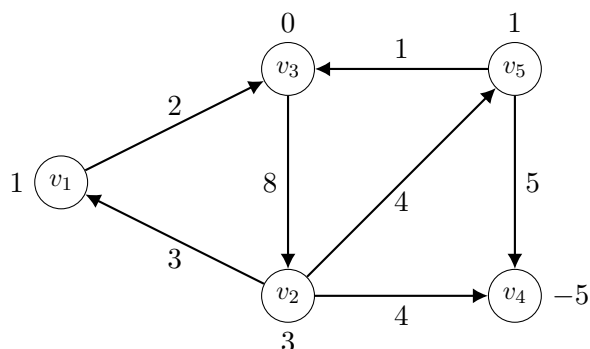
(a) En perfekt matching. (b) En maksimum matching. (c) En maksimal matching.

Figur 2.3

## Kapittel 3

# Nettverk strøm teori

Begrepet nettverk omtales ofte i dagliglivet i forbindelse med infrastrukturer, teknologi, sosiale nettverk osv. I denne oppgaven definerer vi (noe upresist) et *nettverk*  $D = (V, E)$  til å være en rettet graf der nodene og/eller kantene har assosierte tallverdier (typisk; kostnader eller kapasiteter langs kantene, og behov/tilførsel ved nodene) [AMO93, s. 24]. Figur 3.1 viser et eksempel på et nettverk med tall assosiert med både kanter og noder.



Figur 3.1: Et nettverk med tall assosiert med både kanter og noder.

Formålet med å definere et nettverk er å diskutere problemstillinger som lar seg modellere ved at vi bestemmer en strøm i nettverket. Et nettverk kan for eksempel illustrere et veinettverk, med kapasiteter som indikerer hvor mye trafikk det kan gå langs en vei før det blir kø. I en slik kontekst kan det være aktuelt å bestemme hvor mye trafikk det kan gå mellom to destinasjoner, forbundet med flere veier, uten at det blir kø. Trafikkflyten kan da modelleres ved at vi representerer den som en strøm i nettverket og løser problemet som et nettverk strøm problem. Før vi skal beskrive ulike nettverk strøm problemer, introduserer vi i delkapittel 3.1 notasjon og definisjoner hentet fra [Dah13].

### 3.1 Strøm og sirkulasjon

En *strøm* i et nettverk  $D = (V, E)$ , er en funksjon  $x$  som tilordner et reelt tall til hver kant  $e = (i, j) \in E$ . Vi krever vanligvis at strømmen overholder en nedre og øvre kapasitet langs hver kant. Ved å la  $l$  og  $u$  være funksjoner som tildeler kant  $e \in E$  verdier s.a.  $l(e) \leq x(e) \leq u(e)$  kan kapasitetsbegrensningene på strømmen formuleres som:

$$l(e) \leq x(e) \leq u(e) \quad \forall e \in E.$$

### Kapittel 3. Nettverk strøm teori

For en node  $v \in V$  definerer vi mengden kanter som går ut fra  $v$ ,

$$\delta^+(v) = \{e \in E : e = (v, w) \text{ for } w \in V\}$$

og mengden kanter som går inn til  $v$

$$\delta^-(v) = \{e \in E : e = (u, v) \text{ for } u \in V\}.$$

For en strøm  $x$  er *divergensen* av  $x$  i en node  $v$  definert som som differansen mellom den totale strømmen som går ut og inn fra  $v$ , det vil si

$$\operatorname{div}_x(v) = \sum_{e \in \delta^+(v)} x(e) - \sum_{e \in \delta^-(v)} x(e).$$

**Lemma 3.1.1.** *Generelt er*

$$\sum_{v \in V} \operatorname{div}_x(v) = 0.$$

*Bevis.*

$$\sum_{v \in V} \operatorname{div}_x(v) = \sum_{v \in V} \left[ \sum_{e \in \delta^+(v)} x(e) - \sum_{e \in \delta^-(v)} x(e) \right].$$

Hver kant  $e = (i, j) \in E$  er med i både  $\delta^+(j)$  og  $\delta^-(i)$ . Det gir at hvert ledd  $x(i, j)$  er inneholdt i dobbeltsummen med både positivt og negativt fortegn, og derfor kanselleres. ■

Vi er i hovedsak interessert i strøm med en gitt divergens i hver node. Matematisk kan det formuleres som en strøm  $x$  som oppfyller

$$\sum_{e \in \delta^+(v)} x(e) - \sum_{(e) \in \delta^-(v)} x(e) = b(v), \quad v \in V.$$

Dersom divergensen av en strøm  $x$  er lik null i hver node, kalles  $x$  en *sirkulasjon*.

For en delmengde  $S$  av  $V$  definerer vi mengden kanter som går ut fra  $S$ ,

$$\delta^+(S) = \{e \in E : e = (v, w) \text{ for } v \in S, w \notin S\}$$

og mengden kanter som går inn til  $S$ ,

$$\delta^-(S) = \{e \in E : e = (v, w) \text{ for } v \notin S, w \in S\}.$$

**Teorem 3.1.2.** (Hoffmans sirkulasjonsteorem, [Dah13, s. 4]). La  $l, u : E \rightarrow \mathbb{R}$  oppfylle  $l \leq u$ . Da eksisterer det en sirkulasjon  $x$  i  $D$  slik at

$$l \leq x \leq u$$

hvis og bare hvis

$$\sum_{e \in \delta^-(S)} l(e) \leq \sum_{e \in \delta^+(S)} u(e) \quad (S \subseteq V).$$

For bevis, se [Dah13].

## 3.2 Maksimum strøm problemet

Et anvendtbart nettverk strøm problem er det som kalles *maksimum strøm problemet*. Dette problemet innebærer å maksimere en strøm mellom to noder i et nettverk. For å gi en presis formulering, introduserer vi notasjon hentet fra [Dah13].

La  $D = (V, E)$  være et nettverk med en ikke-negativ kapasitetsfunksjon  $c: E \rightarrow \mathbb{R}_+$ . Vi betegner nodene strømmen skal gå mellom med  $s$  og  $t$ , og kaller dem for henholdsvis *kilden* og *sluket* i nettverket. Videre definerer vi en  $st$ -strøm som en strøm  $x$  som oppfyller

$$\sum_{e \in \delta^+(v)} x(e) = \sum_{(e) \in \delta^-(v)} x(e) \quad \forall \quad v \in V \setminus \{s, t\}, \quad (3.1a)$$

$$0 \leq x(e) \leq c(e), \quad \forall \quad e \in E. \quad (3.1b)$$

Verdien av en  $st$ -strøm er definert som den totale strømmen som går ut fra sluket og betegnes med  $\text{val}(x)$ . Matematisk kan verdien av en  $st$ -strøm  $x$  formuleres som

$$\text{val}(x) = \sum_{e \in \delta^+(s)} x(e).$$

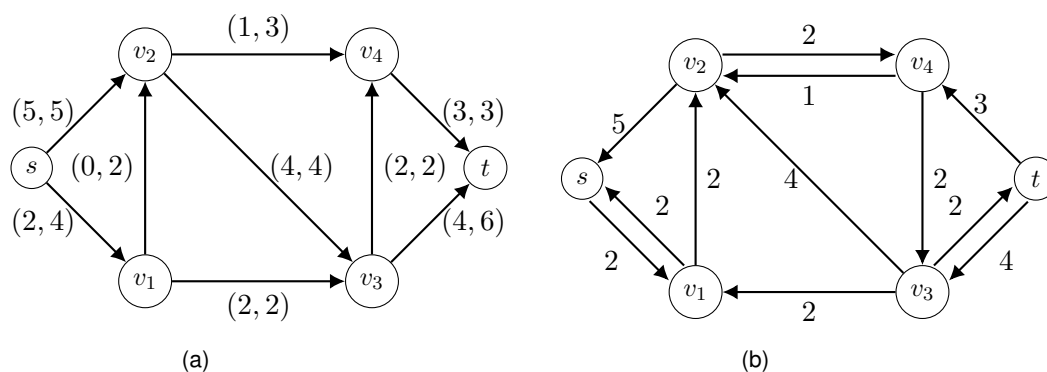
Videre antas det at ingen nettverk inneholder kanter med  $s$  som sluttnode eller  $t$  som startnode. Det impliserer at verdien av hver  $st$ -strøm er lik divergensen av strømmen i node  $s$ , som er lik den negative divergensen av strømmen i node  $t$ , det vil si

$$\text{val}(x) = \text{div}_x(s) = -\text{div}_x(t).$$

Maksimum strøm problemet kan dermed formuleres som å finne en  $st$ -strøm  $x$  som maksimerer  $\text{val}(x)$ . En slik strøm vil videre refereres til som en *maksimum  $st$ -strøm*.

Med formål om å presentere Ford-Fulkerson algoritme for å løse maksimum strøm problemet introduserer vi det [AMO93, s. 177] definerer som *residualnettverk* og *residualkapasiteter*. Gitt et nettverk  $D = (V, E)$  og en strøm  $x$ , defineres residualnettverket,  $D(x)$ , som nettverket vi får ved å erstatte hver kant  $(i, j) \in E$  med fremoverkanten  $(i, j)$  og bakoverkanten  $(j, i)$ . Videre tilordnes begge kantene en ikke-negativ residualkapasitet  $r$ . For kant  $(i, j)$  er residualkapasiteten definert som den ubrukte kapasiteten langs kanten, det vil si  $r(i, j) = c(i, j) - x(i, j)$ , mens for kant  $(j, i)$  er residualkapasiteten definert som strømmen som kan sendes tilbake langs kant  $(j, i)$ , det vil si  $r(j, i) = x(i, j)$ . Figur 3.2a viser et nettverk med en  $st$ -strøm og kapasiteter tilordnet kantene, og Figur 3.2b viser det tilhørende residualnettverket med residualkapasiteter tilordnet kantene.

Som [AMO93, s. 180], kaller vi en rettet vei fra kilden til sluket i et residual-nettverk for en *tilføyende vei* (engelsk: augmenting path). Ideen med å introdusere en tilføyende vei er å se hvordan vi kan endre strømmen langs gitte kanter i et nettverk med formål om å øke den totale  $st$ -strømmen. I Figur 3.2b er  $s - v_1 - v_2 - v_4 - v_3 - t$  en tilføyende vei: ved å legge til 2 enheter strøm langs kantene  $(s, v_1), (v_1, v_2), (v_2, v_4), (v_4, v_3)$  og  $(v_3, t)$  økes verdien til  $st$ -strømmen fra 7 til 9.



Figur 3.2: (a) Et nettverk med strøm og kapasitet tilordnet kantene. (b) Tilhørende residualnettverk med residualkapasiteter tilordnet kantene. Kun kanter med positiv residualkapasitet vises.

Vi er nå klare til å beskrive Ford-Fulkerson algoritme.

### Ford-Fulkerson algoritmen

Beskrivelsen av algoritmen er hentet fra [Dah13, s. 10].

La  $x$  være en  $st$ -strøm i et nettverk  $D = (V, E)$ , med tilhørende residualnettverk  $D(x)$ .

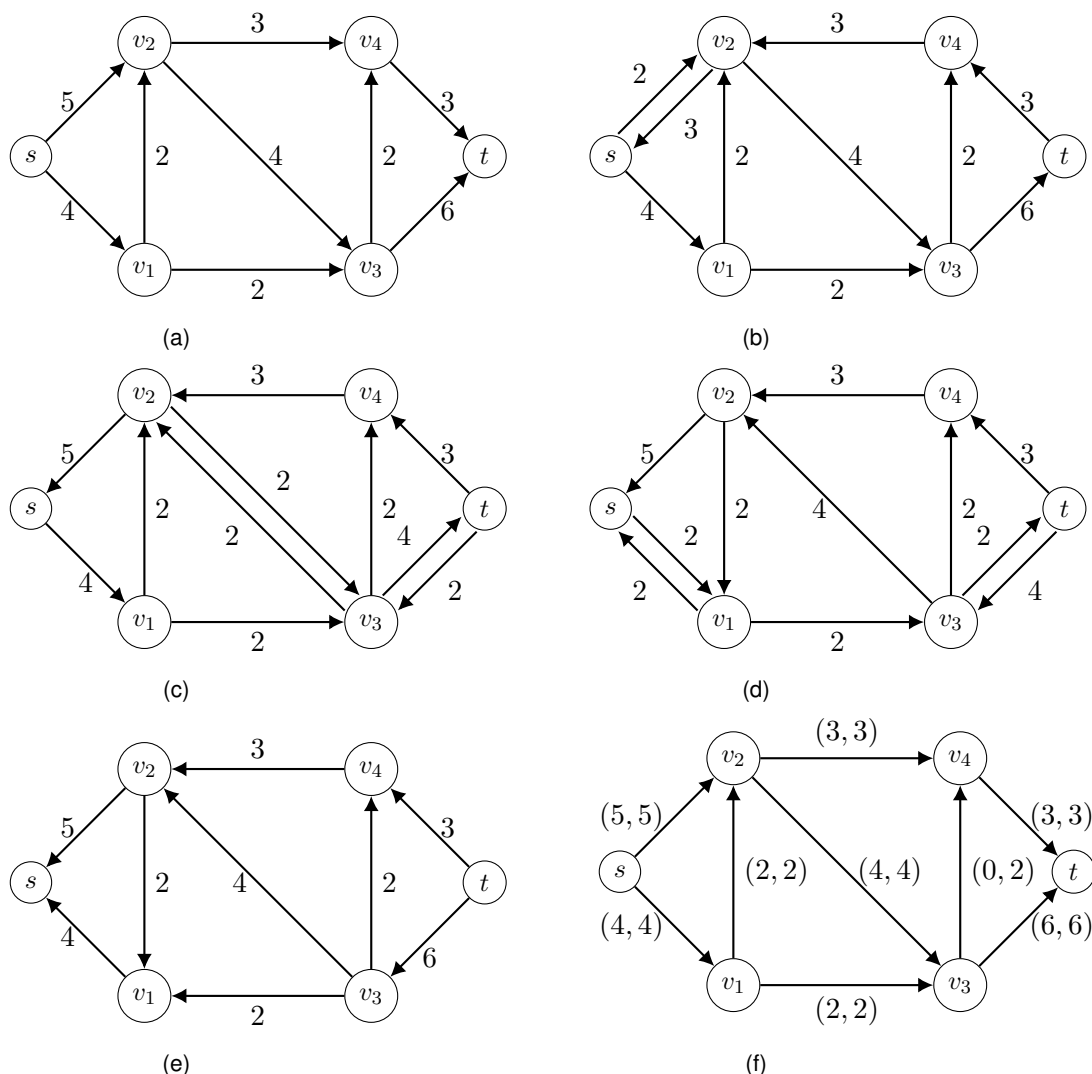
1. Sett  $x := 0$ .
2. Identifiser en tilføyende vei  $P$  fra  $s$  til  $t$  i  $D(x)$ .
  - (a) Hvis en slik vei  $P$  finnes: bestem antall enheter strømmen langs kantene i  $P$  maksimalt kan tilføyes (slik at residualkapasitetene overholdes). Oppdater  $x$  i henhold til dette. Returner til punkt 2.
  - (b) Hvis ingen slik vei  $P$  finnes, så er  $x$  en maksimum  $st$ -strøm.

I slutten av neste delkapittel skal vi bevise punkt 2b av algoritmen.

**Eksempel 3.2.1.** Betrakt nettverket vist i Figur 3.3a.

1. Vi starter med å sette  $x := 0$ . Residualnettverket til  $x$  er da det samme som det innledende nettverket i Figur 3.3a.
2. I Figur 3.3a identifiserer vi veien  $s - v_2 - v_4 - t$  som en tilføyende vei fra  $s$  til  $t$ : langs kantene  $(s, v_2)$ ,  $(v_2, v_4)$  og  $(v_4, t)$  kan  $x$  økes fra 0 til 3. Residualnettverket til  $x$  etter denne økningen er vist i Figur 3.3b.
3. I Figur 3.3b identifiserer vi veien  $s - v_2 - v_3 - t$  som en tilføyende vei fra  $s$  til  $t$ : langs kantene  $(s, v_2)$ ,  $(v_2, v_3)$  og  $(v_3, t)$  kan  $x$  økes med 2. Residualnettverket til  $x$  etter denne økningen er vist i Figur 3.3c.
4. I Figur 3.3c identifiserer vi veien  $s - v_1 - v_2 - v_3 - t$  som en tilføyende vei fra  $s$  til  $t$ : langs kantene  $(s, v_1)$ ,  $(v_1, v_2)$ ,  $(v_2, v_3)$  og  $(v_3, t)$  kan  $x$  økes med 2. Residualnettverket til  $x$  etter denne økningen er vist i Figur 3.3d.
5. I Figur 3.3d identifiserer vi veien  $s - v_1 - v_3 - t$  som en tilføyende vei fra  $s$  til  $t$ : langs kantene  $(s, v_1)$ ,  $(v_1, v_3)$  og  $(v_3, t)$  kan  $x$  økes med 2. Residualnettverket til  $x$  etter denne økningen er vist i Figur 3.3e.

6. I Figur 3.3e er det ingen gjenværende tilføyende vei, og vi konkluderer med at  $x$  er en maksimum  $st$ -strøm. Figur 3.3f viser nettverket med den beregnede maksimum  $st$ -strømmen. Ved å summere opp strømmen som går ut fra node  $s$  får vi  $\text{val}(x) = 9$ .



Figur 3.3: (a) Et nettverk med kapasiteter tilordnet hver kant (residualnettverket for  $x = 0$ ).  
 (b) Residualnettverket etter å ha tilføyet 3 enheter strøm langs veien:  $s - v_2 - v_4 - t$ .  
 (c) Residualnettverket etter å ha tilføyet 2 enheter strøm langs veien:  $s - v_2 - v_3 - t$ .  
 (d) Residualnettverket etter å ha tilføyet 2 enheter strøm langs veien:  $s - v_1 - v_2 - v_3 - t$ .  
 (e) Residualnettverket etter å ha tilføyet 2 enheter strøm langs veien:  $s - v_1 - v_3 - t$ .  
 (f) Nettverket med en maksimum  $st$ -strøm og kapasitet  $(x(e), c(e))$  tilordnet kant  $e$ .

△

**Teorem 3.2.2.** (Heltallsteoremet, [AMO93, s. 186]). Hvis alle kapasiteter i et nettverk er heltall, så har maksimum strøm problemet en heltallsløsning.

*Bevis.* Hentet fra [AMO93, s. 186]. Anta at  $D = (V, E)$  er et nettverk med to noder  $s$  og  $t$ , og en kapasitetsfunksjon  $c : E \rightarrow \mathbb{Z}^+$ . Vi bruker Ford-Fulkerson algoritmen til å bestemme en maksimum  $st$ -strøm i  $D$ . La  $x = 0$ . Identifiser en tilføyende vei i residualnettverket for  $x = 0$  og oppdater  $x$  med minimumet av

residualkapasitetene til kantene langs den tilføyende veien. Siden  $c$  er en heltallsfunksjon, er alle residualkapasitetene heltall, og  $x$  oppdateres med et heltall. Det impliserer videre at residualkapasitetene i residualnettverket til den oppdaterte strømmen er heltall. Dersom det finnes en tilføyende vei i dette residualnettverket, følger samme argumentasjon til at  $x$  igjen oppdateres med et heltall og vi får et nytt residualnettverk med heltalls residualkapasiteter. Ved å gjenta argumentasjon, inntil det ikke finnes noen gjenværende tilføyende vei, får vi at strømmen stadig oppdateres med et heltall, og resultatet følger. ■

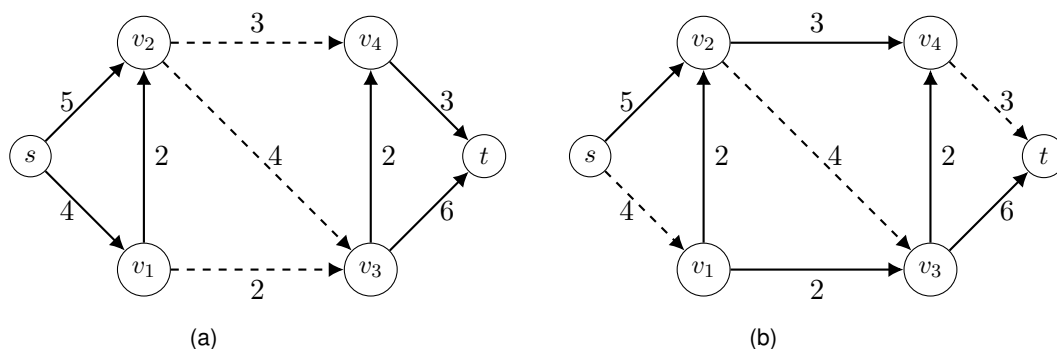
### 3.3 Minimum kutt problemet

Et problem, nært releatert til maksimum strøm problemet, er det som kalles *minimum kutt problemet*. Problemet går ut på å finne et  $st$ -kutt med lavest mulig kapasitet [Dah13, s. 7]. For å forklare hva vi mener med det, innfører vi notasjon hentet fra [AMO93] og [Dah13]. Som tidligere lar vi  $D = (V, E)$  være et nettverk med en ikke-negativ kapasitetsfunksjon  $c: E \rightarrow \mathbb{R}_+$ . Videre deler vi nodemengden  $V$  inn i to ikke-tomme disjunkte mengder  $S$  og  $\bar{S} = V \setminus S$ . Et *kutt* er definert som mengden kanter  $e = (i, j)$  som er slik at  $i \in S$  og  $j \in \bar{S}$ , og betegnes med  $(S, \bar{S})$ . Et  $st$ -kutt er definert som et kutt der node  $s \in S$  og node  $t \in \bar{S}$ .

Tilsvarende som vi definerte verdien av en strøm for å sammenligne ulike  $st$ -strømmer, definerer vi kapasiteten av et kutt for å sammenligne ulike  $st$ -kutt. Kapasiteten av et kutt  $(S, \bar{S})$  er definert som summen av kapasitetene langs kantene som går ut av kuttet og betegnes med  $\text{cap}(S, \bar{S})$ . Matematisk kan kapasiteten av et kutt  $(S, \bar{S})$  formuleres som

$$\text{cap}(S, \bar{S}) = \sum_{e \in (S, \bar{S})} c(e).$$

De stiplede linjene i Figur 3.4 illustrerer to forskjellige  $st$ -kutt i et nettverk med øvre kapasiteter tilordnet kantene.



Figur 3.4: (a)  $S = \{s, v_1, v_2\}$ ,  $\bar{S} = \{v_3, v_4, t\}$ ,  $\text{cap}(S, \bar{S}) = 9$ . (b)  $S = \{s, v_2, v_4\}$ ,  $\bar{S} = \{v_1, v_3, t\}$ ,  $\text{cap}(S, \bar{S}) = 11$ .

Som nevnt innledningsvis innebærer minimum kutt problemet å bestemme et  $st$ -kutt med lavest mulig kapasitet. For å bestemme et slikt kutt, som heretter vil refereres til som et *minimum  $st$ -kutt*, skal vi utnytte at det har en relasjon til det korresponderende maksimum strøm problemet.

**Lemma 3.3.1.** [AMO93, s. 179]. *Verdien av hver  $st$ -strøm i et nettverk, er mindre eller lik kapasiteten av hvert  $st$ -kutt i nettverket.*



*Bevis.* Hentet fra [AMO93, s. 179]. La  $x$  være en  $st$ -strøm i et nettverk  $D = (V, E)$  og la  $(S, \bar{S})$  være et  $st$ -kutt i nettverket. Fra (3.1a) har vi at  $\text{div}_x(v) = 0$  for alle  $v \in V \setminus \{s, t\}$ . Verdien av  $x$  kan derfor uttrykkes som:

$$\text{val}(x) = \sum_{i \in S} \left[ \sum_{e \in \delta^+(i)} x(e) - \sum_{e \in \delta^-(i)} x(e) \right].$$

Ved å skrive ut dobbeltsummen, ser vi at strøm mellom noder i  $S$  er inneholdt med både positivt og negativt fortegn, og kanselleres. Dobbeltsummen kan derfor forenkles til:

$$\text{val}(x) = \sum_{e \in (S, \bar{S})} x(e) - \sum_{e \in (\bar{S}, S)} x(e). \quad (3.2)$$

Ved å erstatte hvert ledd i den første summen i (3.2) med øvre kapasitetsbegrensninger og hvert ledd i den andre summen med 0 får vi:

$$\text{val}(x) \leq \sum_{e \in (S, \bar{S})} c(e) - \sum_{e \in (\bar{S}, S)} 0 = \sum_{e \in (S, \bar{S})} c(e) = \text{cap}(S, \bar{S}). \quad (3.3)$$

■

Fordi Lemma 3.3.1 skal gjelde for hver  $st$ -strøm og hvert  $st$ -kutt, kan vi ta maksimum over verdiene av alle  $st$ -strømmer og minimum over kapasitetene av alle  $st$ -kutt [Dah13, s. 7]. Det gir ulikheten:

$$\max\{\text{val}(x) : x \text{ er en } st\text{-strøm}\} \leq \min\{\text{cap}(S, \bar{S}) : (S, \bar{S}) \text{ er et } st\text{-kutt}\}. \quad (3.4)$$

Følgende teorem gir at det faktisk er likhet.

**Teorem 3.3.2.** (Maks-strøm min-kutt teoremet, [AMO93, s. 185]). Maksimumverdien av en strøm fra node  $s$  til  $t$  i et nettverk med kapasiteter tilordnet kantene, er lik minimumkapasiteten over alle  $st$ -kutt i nettverket.

Teoremet er bevist av Ford og Fulkerson for urettede nettverk, og av Dantzig og Fulkerson for rettede nettverk (som vi betrakter her) [Dah13]. Videre følger et bevis hentet fra [AMO93, s. 184–185]. For et bevis som anvender Hoffmans sirkulasjonsteorem (Teorem 3.1.2), se [Dah13, s. 8].

*Bevis.* La  $x$  være en  $st$ -strøm i et nettverk  $D = (V, E)$  og la  $D(x)$  betegne residualnettverket til  $x$ . La videre  $(S, \bar{S})$  være et  $st$ -kutt, der  $S$  er mengden noder  $v \in V$  slik at det finnes en tilføyende vei fra  $s$  til  $v$  i  $D(x)$ . Fra definisjonen av et  $st$ -kutt følger det at  $s \in S$  og  $t \in \bar{S}$ , som impliserer at det ikke kan være noen tilføyende vei fra  $s$  til  $t$  i  $D(x)$ .

La  $(i, j)$  være en kant i  $D$  slik at  $i \in S$  og  $j \in \bar{S}$ . Da må  $x(i, j) = c(i, j)$ , for hvis  $x(i, j) < c(i, j)$  kunne vi tilføyd strøm langs kanten  $(i, j)$  i  $D(x)$  som impliserer  $j \in S$ . La videre  $(i, j)$  være en kant  $D$  slik at  $i \in \bar{S}$  og  $j \in S$  og. Da må  $x(i, j) = 0$ , for hvis  $x(i, j) > 0$  kunne vi tilføyd strøm langs kanten  $(j, i)$  i  $D(x)$  som impliserer  $i \in S$ .

Dermed får vi:

$$\begin{aligned}
 \text{val}(x) &= \sum_{i \in S} \left[ \sum_{e \in \delta^+(i)} x(e) - \sum_{e \in \delta^-(i)} x(e) \right] \\
 &= \sum_{e \in (S, \bar{S})} x(e) - \sum_{e \in (\bar{S}, S)} x(e) \\
 &= \sum_{e \in (S, \bar{S})} c(e) - \sum_{(e) \in (\bar{S}, S)} 0 \\
 &= \sum_{e \in (S, \bar{S})} c(e) \\
 &= \text{cap}(S, \bar{S}).
 \end{aligned} \tag{3.5}$$

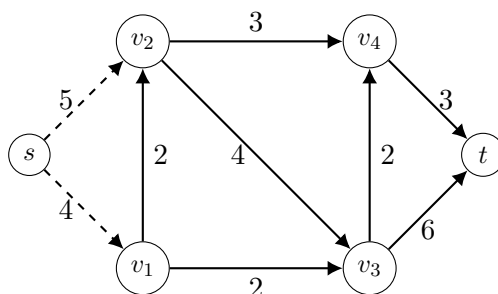
Fordi det er likhet i (3.5) følger det fra (3.4) at  $x$  må være en maksimum  $st$ -strøm og at  $(S, \bar{S})$  er et minimum  $st$ -kutt. ■

Fra beviset av Teorem 3.3.2 følger det i tillegg at dersom det ikke finnes en tilføyende vei fra  $s$  til  $t$  i residualnettverket til  $x$ , så er  $x$  en maksimum  $st$ -strøm. Dette forklarer hvorfor Ford-Fulkerson algoritmen resulterer i en maksimum  $st$ -strøm. Vi kan imidlertid også vise at dersom  $x$  er en maksimum  $st$ -strøm, så finnes det ingen tilføyende vei fra  $s$  til  $t$ . Anta at det finnes en tilføyende vei fra  $s$  til  $t$  i residualnettverket til en  $st$ -strøm  $x$ . Vi kan da øke strømmen langs denne veien og få en  $st$ -strøm med høyere verdi, som impliserer at  $x$  ikke er en maksimum  $st$ -strøm. Vi får dermed følgende proposisjon.

**Proposisjon 3.3.3.** [AMO93, s. 185] *En  $st$ -strøm  $x$  er en maksimum strøm hvis og bare hvis det ikke finnes noen tilføyende vei fra  $s$  til  $t$  i residualnettverket til  $x$ .*

Videre har vi fra beviset av Teorem 3.3.2 at et minimum  $st$ -kutt er gitt som  $(S, \bar{S})$ , der  $S$  er mengden noder som har en tilføyende vei til seg fra  $s$ , i residualnettverket til en maksimum  $st$ -strøm  $x$ . Vi illustrerer relasjonen ved å bestemme et minimum  $st$ -kutt i samme nettverk som vi bestemte en maksimum strøm i, i Eksempel 3.2.1.

**Eksempel 3.3.4.** Betrakt nettverket i Figur 3.3a. Vi utnytter at vi allerede har bestemt en maksimum  $st$ -strøm vist, i Figur 3.3f. Fra residualnettverket til denne strømmen, vist i Figur 3.3e, ser vi at det ikke er noen tilføyende vei fra  $s$  til noen noder. Vi konkluderer derfor med at kuttet med  $S = \{s\}$  og  $\bar{S} = \{v_1, v_2, v_3, v_4, t\}$  definerer et minimum  $st$ -kutt (vist i Figur 3.5). Vi observerer at kapasiteten av dette kuttet er lik 9 som svarer til maksimumverdien av strømmen, bestemt i Eksempel 3.2.1.



Figur 3.5: Et minimum  $st$ -kutt som svarer til maksimum  $st$ -strømmen i Figur 3.3f

△

### 3.4 Det tillatte strøm problemet

Et tredje nettverk strøm problem, er det jeg oversetter til å kalle *det tillatte strøm problemet* (engelsk: The feasible flow problem). Den videre beskrivelsen av dette problemet er basert på [AMO93]. La  $D = (V, E)$  være et nettverk med en kapasitetsfunksjon  $c$  som tilordner hver kant  $e \in E$  en ikke-negativ øvre kapasitet, og en forsyningsfunksjon  $b$  som tilordner hver node  $v \in V$  tilførsel/behov. Det tillatte strøm problemet innebærer å bestemme en strøm  $x$  i  $D$  som oppfyller

$$\sum_{e \in \delta^+(v)} x(e) - \sum_{(e) \in \delta^-(v)} x(e) = b(v), \quad \forall v \in V \quad (3.6)$$

$$0 \leq x(e) \leq c(e) \quad \forall (e) \in E,$$

der vi antar  $\sum_{i \in V} b(i) = 0$ .

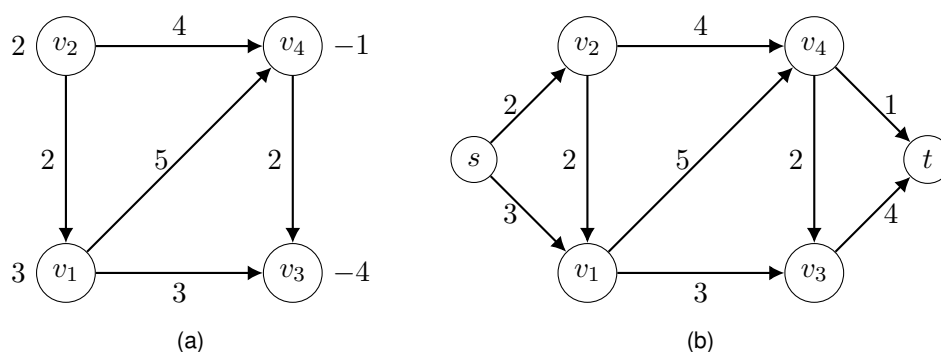
For å løse et tillatt strøm problem innfører vi det vi kaller et *transformert nettverk*. Dette er nettverket vi får ved å utvide  $D$  med to nye noder  $s$  og  $t$ , legge til en kant  $(s, v)$  med kapasitet  $b(v)$  for alle  $v \in V$  med  $b(v) > 0$  og en kant  $(v, t)$  med kapasitet  $-b(v)$  for alle  $v \in V$  med  $b(v) < 0$ .

**Teorem 3.4.1.** [AMO93, s. 170]. La  $D = (V, E)$  være et nettverk med en kapasitetsfunksjon  $c$  og forsyningsfunksjon  $b$ . Hvis  $x$  er en maksimum  $st$ -strøm i det transformerte nettverket, som metter alle kanter fra  $s$  og alle kanter til  $t$  (det vil si at strømmen er lik kapasiteten på disse kantene), så er  $x$  tillatt i  $D$ .

*Bevis.* Hentet fra [AMO93, s. 170]. Anta at  $x$  er en maksimum  $st$ -strøm i det transformerte nettverket som metter alle kanter fra  $s$  og alle kanter til  $t$ . Da er  $x(s, v) = b(v)$  for  $v \in V$  med  $b(v) > 0$  og  $x(v, t) = -b(v)$  for  $v \in V$  med  $b(v) < 0$ , samtidig som  $x$  overholder kapasitetsbegrensningene i det originale nettverket. Dermed oppfyller  $x$  begrenset til det originale nettverket ulikhetene i (3.6). ■

Vi illustrer hvordan vi kan bruke Teorem 3.4.1 til å løse et tillatt strøm problem, gjennom et eksempel.

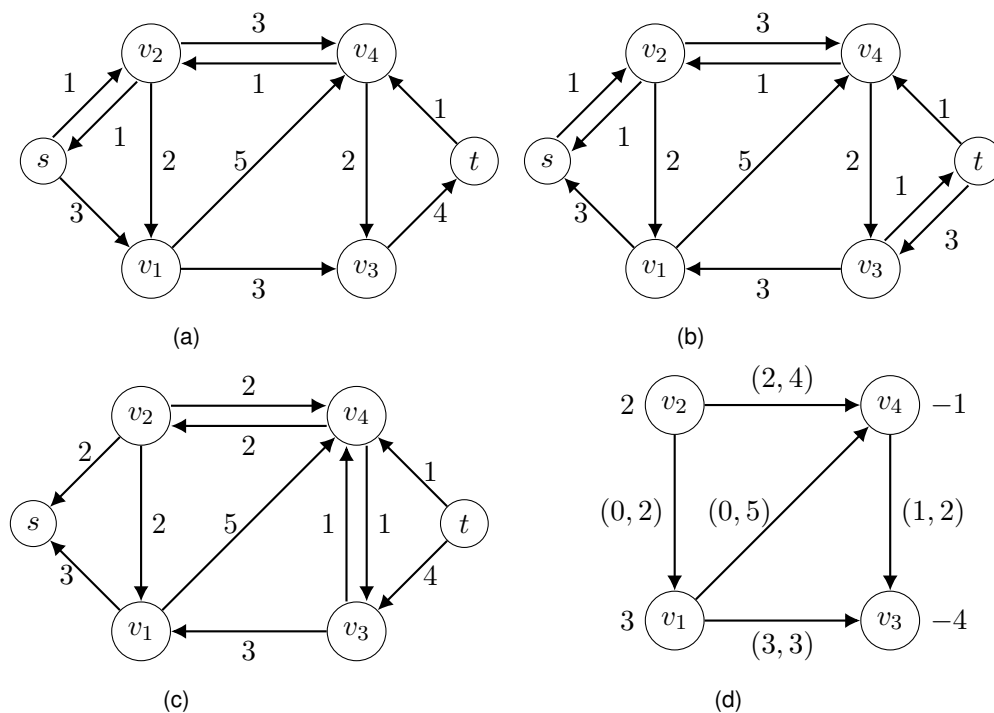
**Eksempel 3.4.2.** Betrakt nettverket vist i Figur 3.6a. Vi vil finne en strøm som overholder kapasitetsbegrensningene langs kantene og oppfyller divergensen i nodene.



Figur 3.6: (a) Et nettverk med kapasiteter tilordnet kantene og tilførsel/behov tilordnet nodene. (b) Tilhørende transformert nettverk (residualnettverket for  $x = 0$ ).

Vi konstruerer først det transformerte nettverket, vist i Figur 3.6b. Videre bruker vi Ford-Fulkerson algoritmen for å bestemme en maksimum  $st$ -strøm i dette nettverket.

1. Vi setter  $x := 0$ . Residualnettverket til  $x$  er da det samme som det innledende nettverket i Figur 3.6b.
2. I Figur 3.6b identifiserer vi veien  $s - v_2 - v_4 - t$  som en tilføyende vei fra  $s$  til  $t$ : langs kantene  $(s, v_2), (v_2, v_4)$  og  $(v_4, t)$  kan  $x$  økes med 1. Residualnettverket til  $x$  etter denne økningen er vist i Figur 3.7a.
3. I Figur 3.7a identifiserer vi veien  $s - v_1 - v_3 - t$  som en tilføyende vei fra  $s$  til  $t$ : langs kantene  $(s, v_1), (v_1, v_3)$  og  $(v_3, t)$  kan  $x$  økes med 3. Residualnettverket til  $x$  etter denne økningen er vist i Figur 3.7b.
4. I Figur 3.7b identifiserer vi veien  $s - v_2 - v_4 - v_3 - t$  som en tilføyende vei fra  $s$  til  $t$ : langs kantene  $(s, v_2), (v_2, v_4), (v_4, v_3)$  og  $(v_3, t)$  kan  $x$  økes med 1. Residualnettverket til  $x$  etter denne økningen er vist i Figur 3.7c.
5. I Figur 3.7c er det ingen gjenværende tilføyende vei, og vi konkluderer med at  $x$  er en maksimum  $st$ -strøm. Fordi  $x$  metter alle kanter fra  $s$  og alle kanter til  $t$ , følger det fra Teorem 3.4.1 at strømmen er tillatt i det originale nettverket. Figur 3.7d viser det originale nettverket med den tillatte strømmen.

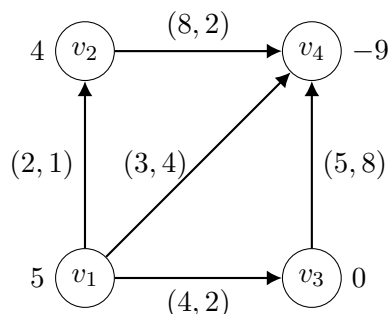


Figur 3.7: (a) Residualnettverket etter å ha tilføyet 1 enhet strøm langs vei  $s - v_2 - v_4 - t$ . (b) Residualnettverket etter å ha tilføyet 3 enheter strøm langs vei  $s - v_1 - v_3 - t$ . (c) Residualnettverket etter å ha tilføyet 1 enhet strøm langs vei  $s - v_2 - v_4 - v_3 - t$ . (d) Det originale nettverket med en tillatt strøm og kapasitet  $(x(e), c(e))$  tilordnet kant  $e$ .

△

### 3.5 Minimum kost strøm problemet

Det siste nettverk strøm problemet vi skal presentere er det som kalles *minimum kost strøm problemet*. Som navnet tilsier innebærer problemet å minimere kostnaden assosiert med en strøm i et nettverk. For at dette skal gi mening må vi assosiere hver kant med en kostnad som indikerer hvor mye det koster å sende én enhet strøm langs denne kanten. I tillegg tilordner vi hver kant en ikke-negativ kapasitetsbegrensning, og hver node tilførsel/behov. Figur 3.8 viser et eksempel på et nettverk med kapasiteter og kostander tilordnet kantene og tilførsel/behov tilordnet nodene.



Figur 3.8: Et nettverk med en kapasitet og en kostnad tilordnet kantene og tilførsel/behov tilordnet nodene.

Den videre beskrivelsen av problemet er basert på [AMO93]. La  $D = (V, E)$  være et nettverk med en kostnadsfunksjon  $k$  som tilordner hver kant  $e \in E$  en kostnad per enhet strøm, en kapasitetsfunksjon  $c$  som tilordner hver kant  $e \in E$  en ikke-negativ øvre kapasitet, og en forsyningsfunksjon  $b$  som tilordner hver node  $v \in V$  tilførsel/behov. Minimum kost strøm problemet innebærer å minimere

$$\sum_{e \in E} k(e) \cdot x(e)$$

Forutsatt

$$\sum_{e \in \delta^+(v)} x(e) - \sum_{(e) \in \delta^-(v)} x(e) = b(v) \quad \forall v \in V, \quad (3.7)$$

$$0 \leq x(e) \leq c(e), \quad \forall e \in E.$$

Som før antar vi

$$\sum_{i \in V} b(i) = 0.$$

En måte å løse minimum kost strøm problemet på, er å benytte det jeg har oversatt til “Syklus-kansellering Algoritmen” (engelsk: Cycle-Canceling Algorithm). Beskrivelsen av algoritmen er hentet fra [AMO93, s. 317].

#### Syklus-kansellering Algoritmen

1. Etabler en tillatt løsning  $x$ . Det gjøres ved å redusere problemet til å finne en strøm  $x$  som oppfyller betingelsene i (3.7). Problemet er da et tillatt strøm problem, og løses på samme måte som beskrevet i delkapittel 3.4.

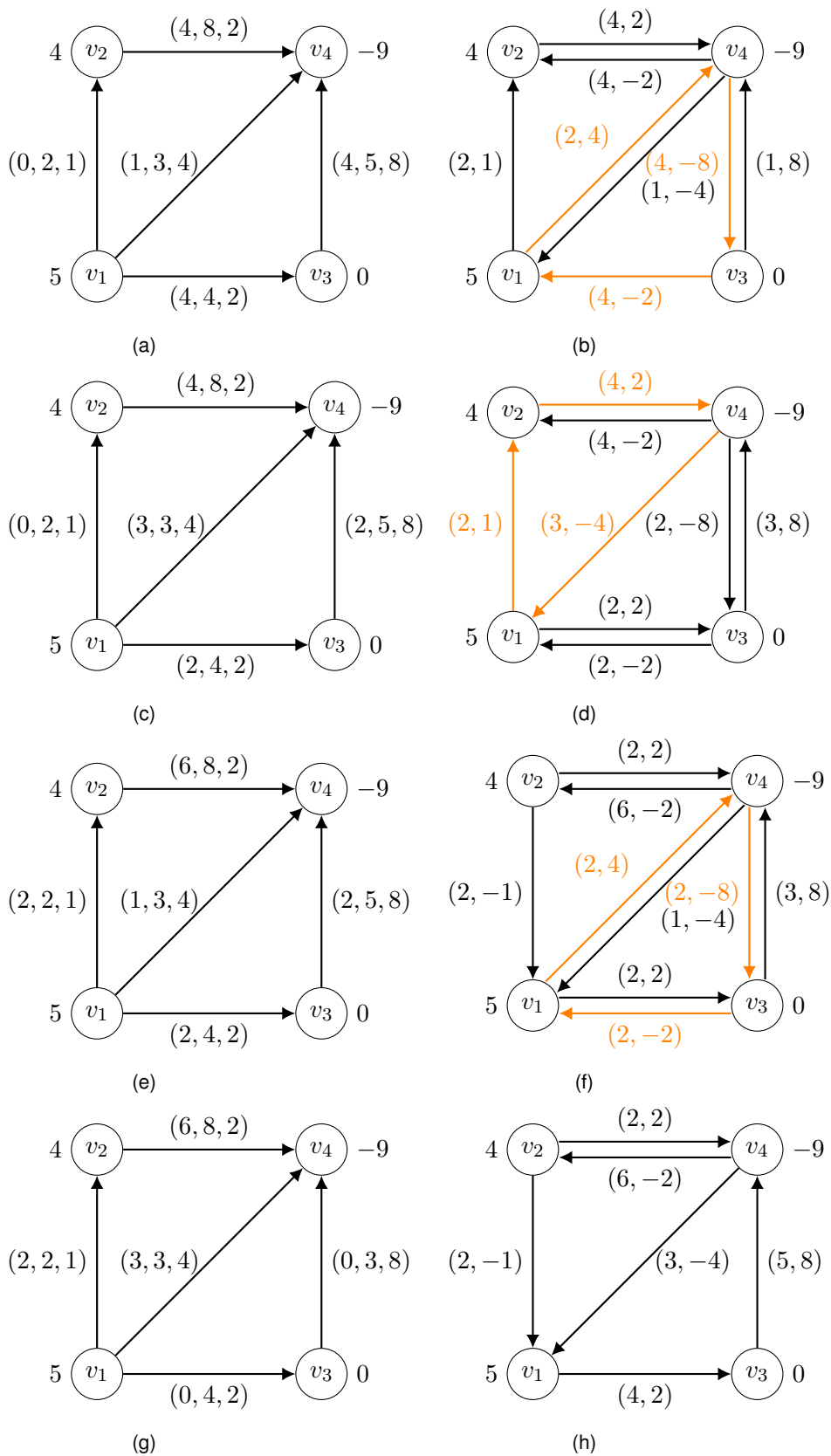
2. Konstruer residualnettverket til  $x$ . Det gjøres på samme måte som tidligere, men i tillegg definerer vi kostnadsfunksjonen på hver bakover-kant til å være den negative verdien av kostnaden på korresponderende fremover-kant. Identifiser deretter en rettet syklus i residualnettverket som er slik at summen av kostnadene langs kantene er negativ.
  - (a) Hvis en slik syklus finnes: Bestem antall enheter strømmen langs kantene i syklusen maksimalt kan tilføyes (slik at residualkapasitetene overholdes). Oppdater  $x$  langs i henhold til dette langs kantene i syklusen. La  $x$  forbli uendret utenfor syklusen. Returner til punkt 2.
  - (b) Hvis ingen slik syklus finnes, så er  $x$  en optimal løsning på minimum kost strøm problemet.

Vi illustrerer algoritmen gjennom et eksempel.

**Eksempel 3.5.1.** Betrakt nettverket vist i Figur 3.8.

1. Vi finner en tillatt strømløsning  $x$  på tilsvarende måte som i Eksempel 3.4.2. Det resulterer i strømmen vist i Figur 3.9a, med kostnad:  $4 \cdot 2 + 1 \cdot 4 + 4 \cdot 2 + 4 \cdot 8 + 0 \cdot 1 = 52$ .
2. Vi konstruerer residualnettverket til  $x$  (se Figur 3.9b) og identifiserer  $v_1 - v_4 - v_3 - v_1$  som en syklus med negativ total kostnad. Vi ser at  $x$  maksimalt kan økes med 2 innad i syklusen. Denne økningen gir strømmen vist i Figur 3.9c, med kostnad:  $4 \cdot 2 + 3 \cdot 4 + 2 \cdot 2 + 2 \cdot 8 + 0 \cdot 1 = 40$ .
3. Vi konstruerer residualnettverket til den oppdaterte strømløsningen (se Figur 3.9d) og identifiserer  $v_1 - v_2 - v_4 - v_1$  som en syklus med negativ total kostnad. Vi ser at  $x$  maksimalt kan økes med 2 innad i syklusen. Denne økningen gir strømmen vist i Figur 3.9e, med kostnad:  $6 \cdot 2 + 1 \cdot 4 + 2 \cdot 2 + 2 \cdot 8 + 2 \cdot 1 = 38$ .
4. Vi konstruerer residualnettverket til den oppdaterte strømløsningen (Figur 3.9f) og identifiserer  $v_1 - v_4 - v_3 - v_1$  som en syklus med negativ total kostnad. Vi ser at  $x$  maksimalt kan økes med 2 innad i syklusen. Denne økningen gir strømmen vist i Figur 3.9g med kostnad:  $6 \cdot 2 + 3 \cdot 4 + 0 \cdot 2 + 0 \cdot 8 + 2 \cdot 1 = 26$ .
5. Vi konstruerer residualnettverket til den oppdaterte strømløsningen, vist i Figur 3.9h, og ser at det ikke er noen sykluser med negativ total kostnad. Vi konkluderer derfor med at  $x$  er en optimal strømløsning, med minimumkostnad lik 26.

△



Figur 3.9: (a), (c), (e) og (g): Nettverket i Figur 3.8 med en tillatt strøm, kapasitet og kostnad  $(x(e), c(e), k(e))$  tilordnet kant  $e$ . (b), (d), (f) og (h): Tilhørende residualnettverk for strømmen i hhv. (a), (c), (e) og (g) med kapasitet og kostnad  $(c(e), k(e))$  tilordnet kant  $e$ .





## Kapittel 4

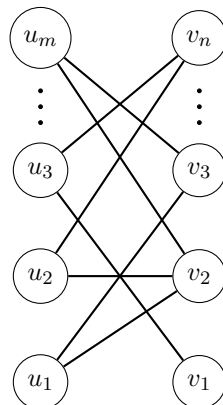
# Anvendelser av nettverk strøm teori

Vi skal i dette kapittelet presentere ulike anvendelser av nettverk strøm teori. Vi skal først beskrive et matchingproblem og se hvordan det kan formuleres og løses som et maksimum strøm problem. Videre skal vi se på sammenhengen mellom strøm i nettverk og transportmatriser. Mer konkret skal vi vise at det å finne en ikke-negativ matrise med gitte linjesummer svarer til å løse et tillatt strøm problem. Til slutt skal vi beskrive et avrundingsproblem for matriser, og vise at også dette kan løses som et tillatt strømproblem.

### 4.1 Matching med maksimum kardinalitet

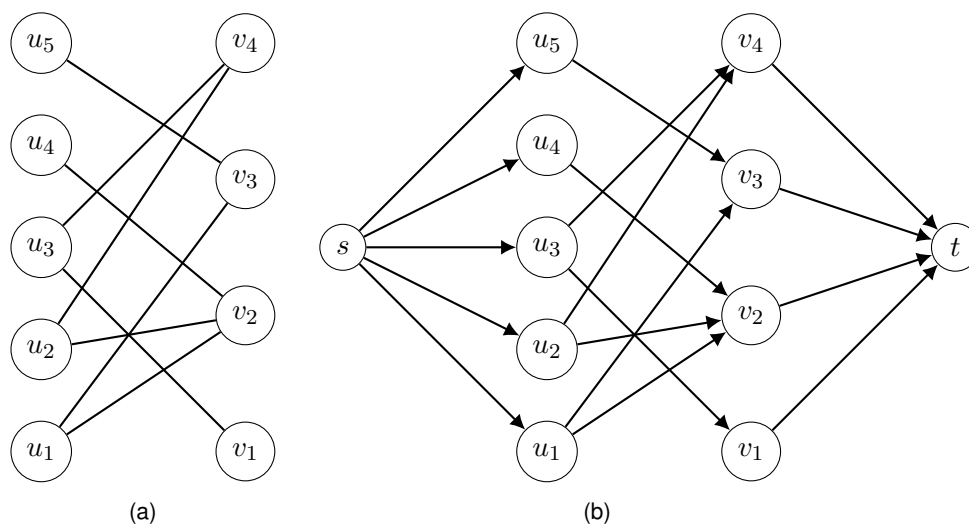
Et anvendbart matchingproblem går ut på å finne en matching som optimerer gitte kriterier [AMO93, s. 9]. Det kan for eksempel innebære å gjøre et laguttak til et stafettlag, der hver løper bare kan tas ut til en etappe han eller hun har trent på. Hvis vi antar at det er  $m$  løpere og  $n$  etapper, der  $n \leq m$ , innebærer et optimalt laguttak å velge ut  $n$  løpere til forskjellige etapper.

**Observasjon 4.1.1.** Vi kan presentere situasjonen beskrevet over med en urettet bipartitt graf  $G = (U \cup V, E)$ , der  $U = (u_1, \dots, u_m)$  og  $V = (v_1, \dots, v_n)$  betegner henholdsvis løper  $i = 1, \dots, m$  og etappe  $j = 1, \dots, n$  og kant  $u_i v_j \in E$  illustrerer at løper  $i$  har trent for etappe  $j$ . Et eksempel på en slik graf er vist i Figur 4.1. Fra denne grafen ser vi at det å gjøre et optimalt laguttak svarer til å finne en matching av kardinalitet  $n$ . Det kan gjøres ved å bestemme en maksimum matching i grafen, og se om den har kardinalitet  $n$ .



Figur 4.1: En urettet bipartitt graf der nodemengdene  $U = (u_1, \dots, u_m)$  og  $V = (v_1, \dots, v_m)$  representerer løpere og etapper, og kantene illustrerer hvilke løpere som er trent for hvilke etapper.

Metoden som presenteres for å bestemme en maksimum matching i en urettet bipartitt graf er hentet fra [AMO93, s. 469], og går ut på at problemet kan løses som et maksimum strøm problem definert på grafens korresponderende nettverk. For å se denne sammenhengen må vi først se hvordan vi gjør om en urettet bipartitt graf til et korresponderende nettverk. La  $G = (U \cup V, E)$  være et urettet bipartitt graf. Erstatt hver urettede kant  $u_i v_j \in E$ , med en rettet kant  $(u_i, v_j)$ . Utvid grafen med to noder  $s$  og  $t$ , og legg til en kant fra  $s$  til hver node  $u \in U$  og en kant fra hver node  $v \in V$  til  $t$ . La til slutt alle kanter ha kapasitet 1. Figur 4.2a viser en urettet bipartitt graf med korresponderende nettverk vist i Figur 4.2b.



Figur 4.2: (a) En urettet bipartitt graf. (b) Det korresponderende nettverket.

**Proposisjon 4.1.2.** [AMO93, s. 469] *Det er 1-1 korrespondanse mellom en matching med kardinalitet  $k$  i en urettet bipartitt graf, og en heltallstrøm med verdi  $k$  i grafens korresponderende nettverket.*

*Bevis.* Hentet fra [AMO93, s. 469–470].

La  $G = (U \cup V, E)$  være en urettet bipartitt graf og anta at  $M = \{(u_1 v_1) \dots, (u_k v_k)\}$  er en matching av kardinalitet  $k$  i  $G$ . Definer en  $st$ -strøm  $x$  i grafens korresponderende nettverk ved å la

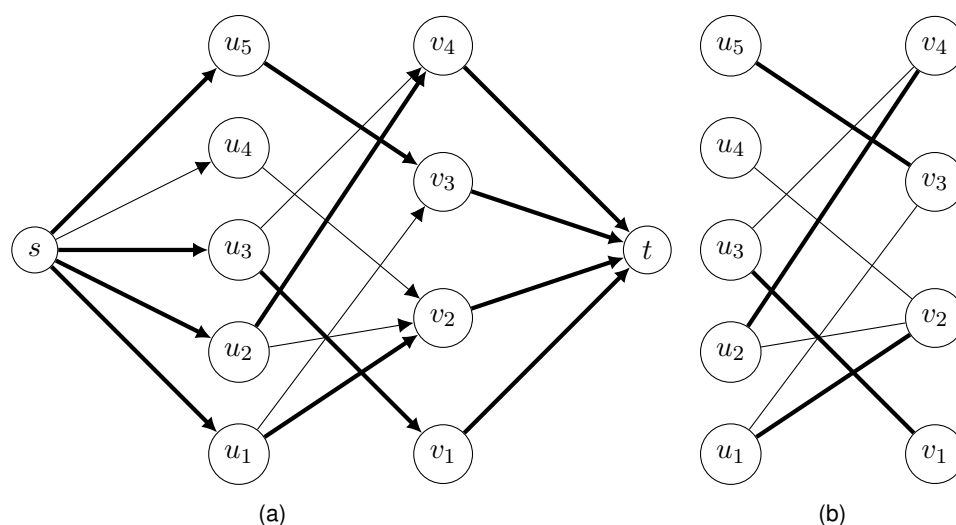
$$x(u_i v_i) = 1, \quad x(s u_i) = 1, \quad x(v_i t) = 1, \quad \forall i \in \{1, \dots, k\}.$$

På alle andre kanter settes strømmen lik 0. Fordi det er nøyaktig  $k$  kanter fra node  $s$  med strøm lik 1 er verdien av  $st$ -strømmen lik  $k$ .

For å vise den andre veien, anta at  $x$  er en  $st$ -strøm i det korresponderende nettverket til en bipartitt graf  $G = (U \cup V, E)$ . Fordi  $\delta^-(u_i) = 1$  for alle  $u_i \in U$ , går det maksimalt strøm langs én kant inn til hver node i  $U$ . For å overholde (3.1a) går det også maksimalt strøm langs én kant ut fra hver node i  $U$ . Tilsvarende er  $\delta^+(v_i) = 1$  for alle  $v_j \in V$ , som impliserer at det maksimalt går strøm langs én kant fra hver node i  $V$ . For å overholde (3.1a) går det også maksimalt strøm langs én kant inn til hver node i  $V$ . Hvis vi ser bort i fra kantene fra  $s$  og kantene til  $t$ , er derfor hver node i  $U \cup V$  maksimalt inntil en kant det går strøm langs. Det impliserer at  $x$  begrenset til  $E$  definerer en matching i  $G$ . Hvis verdien av  $x$  er lik  $k \in \mathbb{Z}$ , går det strøm langs  $k$  kanter mellom  $U$  og  $V$ , som svarer til at matchingen har kardinalitet  $k$ . ■

Fra Proposisjon 4.1.2 følger det at vi kan bestemme en maksimum matching i en bipartitt graf ved å bestemme en maksimum  $st$ -strøm i grafens korresponderende nettverk. Vi illustrerer fremgangsmåten gjennom et eksempel.

**Eksempel 4.1.3.** Vi skal bestemme en maksimum matching i grafen vist i Figur 4.2a ved å finne en maksimum  $st$ -strøm i grafens korresponderende nettverk, vist i Figur 4.2b. Som før bruker vi Ford-Fulkerson algoritmen til å bestemme en maksimum  $st$ -strøm. Algoritmen resulterer i  $st$ -strømmen vist i Figur 4.3a, med verdi 4. Denne strømmen svarer til matchingen vist i Figur 4.3b med kardinalitet 4. Dersom grafen representerer laguttak-situasjonen presentert innledningsvis, svarer matchingen til et optimalt laguttak, der løper 1 tildeles 2. etappe, løper 2 tildeles 4. etappe, løper 3 tildeles 1. etappe og løper 5 tildeles 3. etappe.  $\triangle$



Figur 4.3: (a) En maksimum  $st$ -strøm illustrert med uthevede kanter. (b) Matchingen som korresponderer med  $st$ -strømmen.

## 4.2 Transportmatriser

Vi skal i dette delkapittelet anvende nettverk strøm teori for å løse det som kalles et *transportproblem* [AMO93, s. 7]. Anta at vi ønsker å kartlegge transport av varer mellom  $m$  ulike varehus og  $n$  ulike butikker, der det skal transporteres  $r_i$  varer fra varehus nummer  $i$  og  $s_j$  varer til varehus nummer  $j$ . Dersom vi antar

$$\sum_{i=1}^m r_i = \sum_{j=1}^n s_j,$$

tilsvarer det å bestemme en varestrøm som oppfyller varehusens beholdninger og butikkens behov, med å finne en ikke-negativ matrise  $A = [a_{ij}]$  med radsumvektor  $R = (r_1, \dots, r_m)$  og kolonnesumvektor  $S = (s_1, \dots, s_n)$ . Som [Dah08], kaller vi en slik matrise for en *transportmatrise* (engelsk: transportation matrices).

Dersom det ikke kreves at matriseelementene skal overholde gitte begrensninger, kan et slikt problem løses ved å anvende det jeg oversetter til “Nordvest-hjørne regelen” (engelsk: North West Corner rule). Beskrivelsen av regelen er hentet fra [Dah08, s. 1845].

### Nordvest-hjørne regelen

Konstruer en tom  $m \times n$  matrise.

$$\begin{array}{cccccc} & & & & & r_1 \\ & & & & & r_2 \\ & & & & & \vdots \\ & & & & & r_m \\ s_1 & s_2 & s_3 & \dots & s_n & \end{array}$$

1. Hvis  $r_1 \leq s_1$ : La  $a_{11} = r_1$  og  $a_{12} = \dots = a_{1n} = 0$ . Oppdater  $s_1$  ved å trekke fra  $r_1$  og oppdater  $R$  ved å fjerne  $r_1$ .  
Hvis  $r_1 > s_1$ : La  $a_{11} = s_1$  og  $a_{21} = \dots = a_{m1} = 0$ . Oppdater  $r_1$  ved å trekke fra  $s_1$  og oppdater  $S$  ved å fjerne  $s_1$ .
2. Gjenta punkt 1. med oppdaterte vektorer  $R$  og  $S$  til den gjenstående delmatrisen er tom.

Vi illustrerer regelen gjennom et eksempel.

**Eksempel 4.2.1.** La  $R = (10, 8, 9, 11)$  og  $S = (12, 7, 5, 8, 6)$ . Vi konstruerer først en tom  $4 \times 5$  matrise. Videre anvender vi regelen med stadig oppdatert radsunvektor og kolonnesumvektor, og får delmatrisene vist nedenfor.

$$\begin{array}{ccc} \begin{array}{c} \left[ \begin{array}{ccccc} & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \end{array} \right] & \begin{array}{l} 10 \\ 8 \\ 9 \\ 11 \end{array} \\ 12 \quad 7 \quad 5 \quad 8 \quad 6 \end{array} & \begin{array}{c} \left[ \begin{array}{ccccc} 10 & 0 & 0 & 0 & 0 \\ & & & & \\ & & & & \\ & & & & \\ & & & & \end{array} \right] & \begin{array}{l} 8 \\ 9 \\ 11 \end{array} \\ 2 \quad 7 \quad 5 \quad 8 \quad 6 \end{array} & \begin{array}{c} \left[ \begin{array}{ccccc} 10 & 0 & 0 & 0 & 0 \\ 2 & & & & \\ 0 & & & & \\ 0 & & & & \end{array} \right] & \begin{array}{l} 6 \\ 9 \\ 11 \end{array} \\ 7 \quad 5 \quad 8 \quad 6 \end{array} \end{array}$$

$$\begin{array}{ccc} \begin{array}{c} \left[ \begin{array}{ccccc} 10 & 0 & 0 & 0 & 0 \\ 2 & 6 & 0 & 0 & 0 \\ 0 & & & & \\ 0 & & & & \end{array} \right] & \begin{array}{l} 9 \\ 11 \end{array} \\ 1 \quad 5 \quad 8 \quad 6 \end{array} & \begin{array}{c} \left[ \begin{array}{ccccc} 10 & 0 & 0 & 0 & 0 \\ 2 & 6 & 0 & 0 & 0 \\ 0 & 1 & & & \\ 0 & 0 & & & \end{array} \right] & \begin{array}{l} 8 \\ 11 \end{array} \\ 5 \quad 8 \quad 6 \end{array} & \begin{array}{c} \left[ \begin{array}{ccccc} 10 & 0 & 0 & 0 & 0 \\ 2 & 6 & 0 & 0 & 0 \\ 0 & 1 & 5 & & \\ 0 & 0 & 0 & & \end{array} \right] & \begin{array}{l} 3 \\ 11 \end{array} \\ 8 \quad 6 \end{array} \end{array}$$

$$\begin{array}{ccc} \begin{array}{c} \left[ \begin{array}{ccccc} 10 & 0 & 0 & 0 & 0 \\ 2 & 6 & 0 & 0 & 0 \\ 0 & 1 & 5 & 3 & 0 \\ 0 & 0 & 0 & & \end{array} \right] & \begin{array}{l} 11 \end{array} \\ 5 \quad 6 \end{array} & \begin{array}{c} \left[ \begin{array}{ccccc} 10 & 0 & 0 & 0 & 0 \\ 2 & 6 & 0 & 0 & 0 \\ 0 & 1 & 5 & 3 & 0 \\ 0 & 0 & 0 & 5 & \end{array} \right] & \begin{array}{l} 6 \\ 6 \end{array} \end{array} & \begin{array}{c} \left[ \begin{array}{ccccc} 10 & 0 & 0 & 0 & 0 \\ 2 & 6 & 0 & 0 & 0 \\ 0 & 1 & 5 & 3 & 0 \\ 0 & 0 & 0 & 5 & 6 \end{array} \right] & \end{array} \end{array}$$

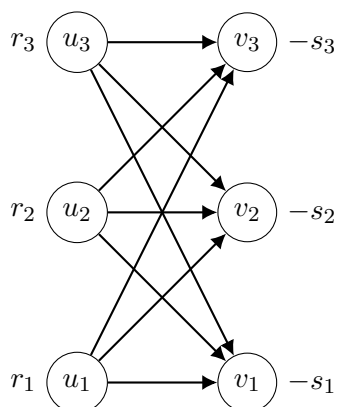
△

Videre skal vi anta at det er begrensinger på antall varer som kan sendes på hver vei mellom produsent og butikk, noe som innebærer at matriseelementene må overholde øvre begrensninger. For å løse dette problemet, kan vi ikke lenger bruke Nordvest-hjørne regelen. Vi skal heller representere matrisen med et nettverk, og bestemme en tillatt strøm i dette nettverket.

## Sammenheng mellom transportmatriser og nettverk

La  $A = [a_{ij}]$  være en ikke-negativ  $m \times n$  matrise med radsumvektor  $R = (r_1, r_2, \dots, r_m)$  og kolonnesumvektor  $S = (s_1, s_2, \dots, s_n)$ . Med notasjon hentet fra [Dah13] definerer vi det korresponderende nettverket til  $A$  som den bipartitte grafen med nodemengder  $U = \{u_1, u_2, \dots, u_m\}$  og  $V = \{v_1, v_2, \dots, v_n\}$  assosiert med radene og kolonnene i matrisen, og en kant  $(u_i, v_j)$  for hvert element  $a_{ij}$  i matrisen. Videre tilordner vi hver node  $u_i \in U$  tilførsel  $b(u_i) = r_i$  og hver node  $v_j \in V$  behov  $b(v_j) = -s_j$ . Hvis matriseelement  $a_{ij}$  skal være øvre begrenset av et ikke-negativt tall  $c_{ij}$ , tilordnes kant  $(u_i, v_j)$  øvre kapasitet  $c(u_i, v_j) = c_{ij}$ . Om ikke, tilordnes kant  $(u_i, v_j)$  kapasitet lik minimumet av  $r_i$  og  $s_j$ .

Figur 4.4 viser det korresponderende nettverket til en  $3 \times 3$  matrise med radsumvektor  $R = (r_1, r_2, r_3)$  og kolonnesumvektor  $S = (s_1, s_2, s_3)$ .



Figur 4.4: Det korresponderende nettverket til en  $3 \times 3$  transportmatrise med radsumvektor  $R = (r_1, r_2, r_3)$  og kolonnesumvektor  $S = (s_1, s_2, s_3)$ .

**Observasjon 4.2.2.** (Hentet fra beviset av “Teorem 2.3” i [Dah13]). La  $x$  være en strøm i det korresponderende nettverket til en ikke-negativ  $m \times n$  matrise  $A = [a_{ij}]$ . La videre  $\text{div}_x = b$ . Ved å la  $x(u_i, v_j) = a_{ij}$  ser vi fra (4.1) at strømmen  $x$  tilsvarer at  $A$  har radsumvektor  $R$  og kolonnesumvektor  $S$ .

$$\begin{aligned} \sum_{j=1}^n a_{ij} &= \sum_{j=1}^n x(u_i, v_j) = \text{div}_x(u_i) = b(u_i) = r_i \quad \forall i = 1, 2, \dots, m, \\ \sum_{i=1}^m a_{ij} &= \sum_{i=1}^m x(u_i, v_j) = -\text{div}_x(v_j) = -b(v_j) = s_j \quad \forall j = 1, 2, \dots, n. \end{aligned} \tag{4.1}$$

Fra denne observasjonen får vi følgende korrespondanse: Det å bestemme en transportmatrise med radsumvektor  $R = (r_1, r_2, \dots, r_m)$  og kolonnesumvektor  $S = (s_1, s_2, \dots, s_n)$ , der hvert matriseelement er tilordnet en øvre kapasitet  $c_{ij}$  og vi antar

$$\sum_{i=1}^m r_i = \sum_{j=1}^n s_j,$$

tilsvarende å finne en strøm i matrisens korresponderende nettverk som oppfyller

$$\sum_{j=1}^n x(u_i, v_j) = r_i \quad \forall \quad i = 1, 2, \dots, m,$$

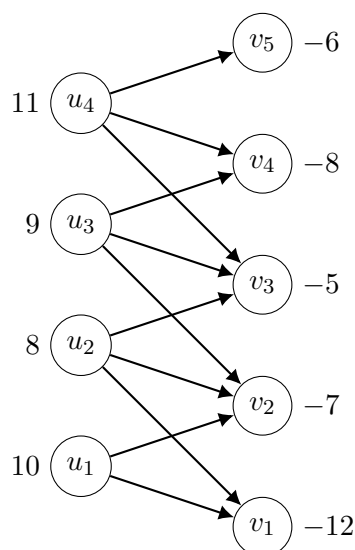
$$\sum_{i=1}^m x(u_i, v_j) = s_j \quad \forall \quad j = 1, 2, \dots, n,$$

$$0 \leq x(u_i, v_j) \leq c(u_i, v_j) = c_{ij} \quad \forall \quad i = 1, 2, \dots, m \text{ og } j = 1, 2, \dots, n.$$

Dette gjenkjenner vi som et tillatt strøm problem definert på matrisens korresponderende nettverk. Som beskrevet i delkapittel 3.4 løser vi problemet ved å bestemme en maksimum  $st$ -strøm i det transformerte nettverket. Dersom  $R$  og  $S$  er heltallsvektorer og kapasitetsbegrensningene er heltall, følger det fra beviset av Helltallssteoremet (Teorem 3.2.2) at Ford-Fulkerson algoritmen resulterer i en heltalls maksimum  $st$ -strøm. Hvis denne strømmen er tillatt i det originale nettverket, korresponderer strømmen med en ikke-negativ heltallsmatrise med radsumvektor  $R$  og kolonnesumvektor  $S$ .

**Eksempel 4.2.3.** La  $R = (10, 8, 9, 11)$  og  $S = (12, 7, 5, 8, 6)$ . La videre element  $c_{ij} = 0$  for  $\{i, j : |i - j| \geq 2\}$ . I et transportproblem, kan slike begrensninger innebære at det kun kan transporteres varer fra et varehus til butikker i nærheten av varehuset.

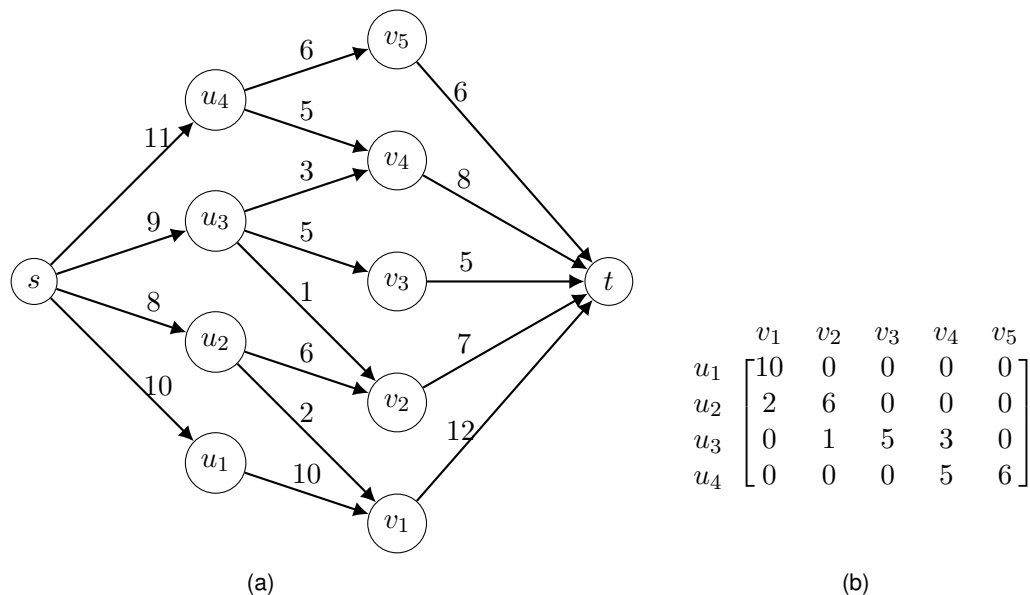
For å finne den tilhørende transportmatrisen, starter vi med å konstruere matrisens korresponderende nettverk. Vi tilordner radnode  $u_i$  tilførsel  $r_i$  og kolonnenode  $v_j$  behov  $-s_j$ . Videre tilordner vi kant  $(i, j)$  øvre kapasitet null for  $\{i, j : |i - j| \geq 2\}$  og  $\min(r_i, s_j)$  ellers. Det resulterer i nettverket vist i Figur 4.5, der kun kanter med øvre kapasitet større enn 0 er tegnet inn.



Figur 4.5: Det korresponderende nettverket til en  $4 \times 5$  transportmatrise med linjesummer  $R = (10, 8, 9, 11)$  og  $S = (12, 7, 5, 8, 6)$ . Kun kanter med kapasitet større enn 0 er tegnet inn.

Videre utvider vi nettverket med to noder  $s$  og  $t$ , legger til en kant  $(s, u_i)$  for  $i = 1, 2, 3, 4$  med kapasitet lik henholdvis 10, 8, 9 og 11, og en kant  $(v_j, t)$  for  $j = 1, 2, 3, 4, 5$  med kapasitet lik henholdvis 12, 7, 5, 8 og 6. For å bestemme en maksimum  $st$ -strøm i dette nettverket bruker vi Ford Fulkerson algoritmen. Det resulterer i  $st$ -strømmen

vist Figur 4.6a, der kun kanter det går strøm langs er tegnet inn. Fordi strømmen metter alle kanter fra  $s$  og alle kanter til  $t$  følger det fra Teorem 3.4.1 at strømmen er tillatt i det originale nettverket. Matrisen som korresponderer med den tillatte strømmen er vist i Figur 4.6b.  $\triangle$



Figur 4.6: (a) Det transformerte nettverket til nettverket i Figur 4.5 med en maksimum  $st$ -strøm  $x$  tilordnet hver kant. (b) Matrisen som korresponderer med strømmen  $x$ .

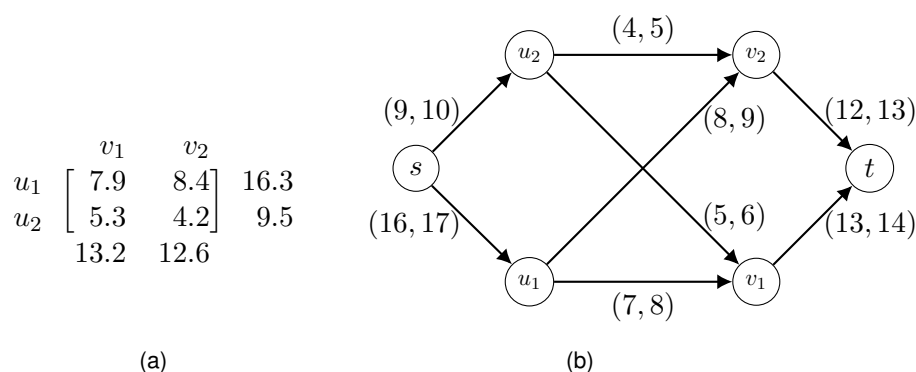
### 4.3 Matriseavrundingsproblemet

I dette delkapittelet skal vi studere det jeg oversetter til å kalle *matriseavrundingsproblemet* (engelsk: matrix rounding problem). Problemet innebærer at vi ønsker å runde av hvert element i en matrise på en slik måte at summen av avrundede elementer i hver rad er lik avrundet radsum, og summen av avrundede elementer i hver kolonne er lik avrundet kolonnesum [AMO93, s. 171]. Vi skal videre se hvordan vi kan formulere og løse problemet som et nettverk strøm problem. Beskrivelsen som presenteres for å konstruere matrisens korresponderende nettverk er hentet fra [AMO93, s. 171–172]. Videre er fremgangsmåten som presenteres for å løse nettverk strøm problemet, basert på metoden som beskrives i [AMO93, s. 193–194] for å bestemme en maksimum strøm i et nettverk med nedre og øvre kapasiteter. Vi trenger imidlertid bare å finne en tillatt strøm for å løse matriseavrundingsproblemet, og metoden som presenteres er derfor avgrenset til å finne en tillatt strøm, og ikke en maksimum strøm.

La  $A = [a_{ij}]$  være en ikke-negativ reell  $m \times n$  matrise. Det første vi gjør er å konstruere matrisens korresponderende nettverk. Vi tar utgangspunkt i den bipartitte grafen med nodemengder  $U = \{u_1, u_2, \dots, u_m\}$  og  $V = \{v_1, v_2, \dots, v_n\}$  assosiert med radene og kolonnene i matrisen, og en kant  $(u_i, v_j)$  for hvert matriseelement  $a_{ij}$ . Deretter utvider vi grafen med nodene  $s$  og  $t$  og legger til en kant  $(s, u_i)$  for hver  $u_i \in U$  og en kant  $(t, v_j)$  for hver  $v_j \in V$ . Videre tilordner vi kant  $(u_i, v_j)$  en nedre og øvre kapasitet som svarer til de to heltallene element  $a_{ij}$  ligger mellom. Tilsvarende tilordner vi kant  $(s, u_i)$  kapasiteter som svarer til de to heltallene radsummen  $\sum_{j=1}^n a_{ij}$  ligger mellom, og kant

$(v_j, t)$  kapasiteter som svarer til de to heltallene kolonnesummen  $\sum_{i=1}^m a_{ij}$  ligger mellom. Vi betegner nettverkets nodemengde  $(U \cup V \cup \{s, t\})$  med  $N$  og kantmengde med  $E$ .

Figur 4.7a viser en  $2 \times 2$  matrise med tilhørende rad- og kolonnesummer og Figur 4.7b viser det korresponderende nettverket.



Figur 4.7: a) En ikke-negativ reell  $2 \times 2$  matrise med tilhørende rad- og kolonnesummer. b) Matrisens korresponderende nettverk. Hver kant er tilordnet et par av en nedre og øvre kapasitet.

**Observasjon 4.3.1.** Det å finne en løsning på et matriseavrundingsproblem svarer til å finne en strøm i det korresponderende nettverket som både oppfyller de nedre og øvre kapasitetsbegrensingene.

For å finne en løsning  $x$  som oppfyller de nedre og øvre kapasitetsbegrensingene introduserer vi et sirkulasjonsproblem. Det går ut på at vi legger til en kant  $(t, s)$  med 0 som nedre og  $\infty$  som øvre kapasitet, og ønsker å identifisere en strøm  $x$  som oppfyller betingelsene

$$\sum_{e \in \delta^+(n)} x(e) - \sum_{e \in \delta^-(n)} x(e) = 0 \quad \forall n \in N, \tag{4.2}$$

$$0 \leq l(e) \leq x(e) \leq c(e) \quad \forall e \in E \cup (t, s),$$

der  $l(e)$  og  $c(e)$  betegner henholdsvis nedre og øvre kapasitet assosiert med kant  $e$ .

Fra Hoffmans sirkulasjonsteorem (Teorem 3.1.2) følger det at det eksisterer en løsning  $x$  som oppfyller (4.2) hvis og bare hvis

$$\sum_{e \in \delta^-(S)} l(e) \leq \sum_{e \in \delta^+(S)} u(e) \quad (S \subseteq N). \tag{4.3}$$

Fordi det er vanskelig å se når (4.3) er oppfylt, gjør vi heller problemet om til et tillatt strøm problem, og bruker Teorem 3.4.1 for å avgjøre om det finnes en løsning. For å få problemet på samme form som i (3.6) fjerner vi midlertidig de nedre kapasitetene. Det gjør vi ved å definere en ny variabel  $\tilde{x}$  som

$$\tilde{x}(e) = x(e) - l(e) \quad \forall e \in E \cup (t, s).$$

Ved å erstatte  $x(e)$  med  $\tilde{x}(e) + l(e)$  i (4.2) omformuleres problemet til å bestemme en



tillatt løsning  $\tilde{x}$  som oppfyller betingelsene

$$\sum_{e \in \delta^+(n)} \tilde{x}(e) - \sum_{e \in \delta^-(n)} \tilde{x}(e) = b(n) \quad \forall n \in N, \quad (4.4)$$

$$0 \leq \tilde{x}(e) \leq u(e) - l(e) \quad \forall e \in E \cup (t, s),$$

der vi definerer

$$b(n) = \sum_{e \in \delta^-(n)} l(e) - \sum_{e \in \delta^+(n)} l(e).$$

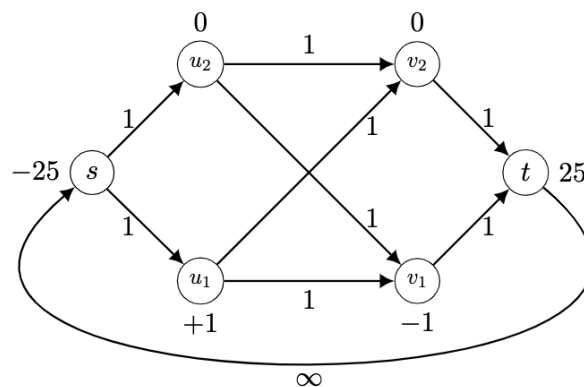
For å oppfylle antakelsen i (3.6) antar vi  $\sum_{n \in N} b(n) = 0$ .

For å løse det tillatte strømproblemet, utvider vi nettverket med to noder  $s'$  og  $t'$ , legger til en kant  $(s', n)$  med kapasitet  $b(n)$  for alle  $n \in N$  med  $b(n) > 0$  og en kant  $(n, t')$  med kapasitet  $-b(n)$  for alle  $n \in N$  med  $b(n) < 0$ . Videre bestemmer vi en maksimum  $s't'$ -strøm (med hensyn på  $\tilde{x}$ ) i dette nettverket. Dersom maksimum  $s't'$ -strømmen metter alle kanter fra  $s'$  og alle kanter til  $t'$  følger det fra Teorem 3.4.1 at strømmen er tillatt i nettverket uten nodene  $s'$  og  $t'$ . Dersom problemet har en tillatt løsning  $\tilde{x}$ , settes den inn i ligningene

$$x(e) = \tilde{x}(e) + l(e) \quad \forall e \in E, \quad (4.5)$$

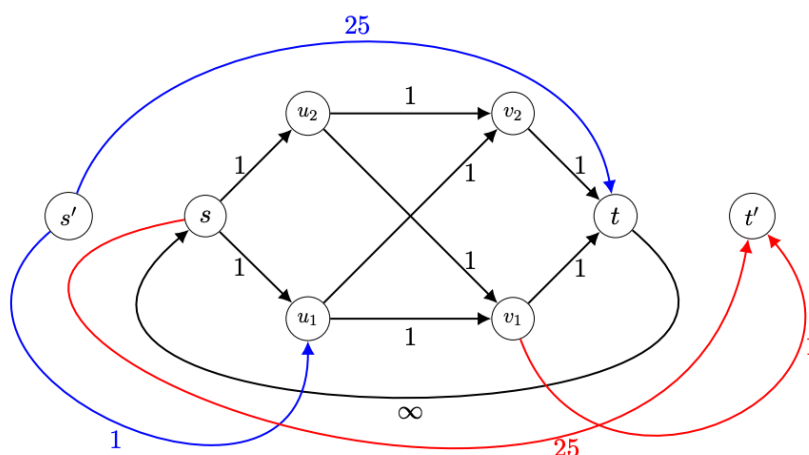
og vi får en tillatt løsning  $x$  som oppfyller betingelsene i (4.2). Til slutt fjerner vi kanten fra  $t$  til  $s$ .

**Eksempel 4.3.2.** Vi skal løse avrundingsproblemet for matrisen vist i Figur 4.7a. Betrakt matrisens korresponderende nettverk, vist i Figur 4.7b. Det første vi gjør er å legge til en kant  $(t, s)$  med 0 som nedre og  $\infty$  som øvre kapasitet. Videre gjør vi et variabelskifte fra  $x$  til  $\tilde{x} = x - l$ , og oppdaterer kapasiteter og tilførsel/behov som beskrevet i (4.4). Det resulterer i nettverket vist i Figur 4.8.



Figur 4.8: Nettverket fra Figur 4.7b med en ekstra kant  $(t, s)$  og etter variabelskifte.

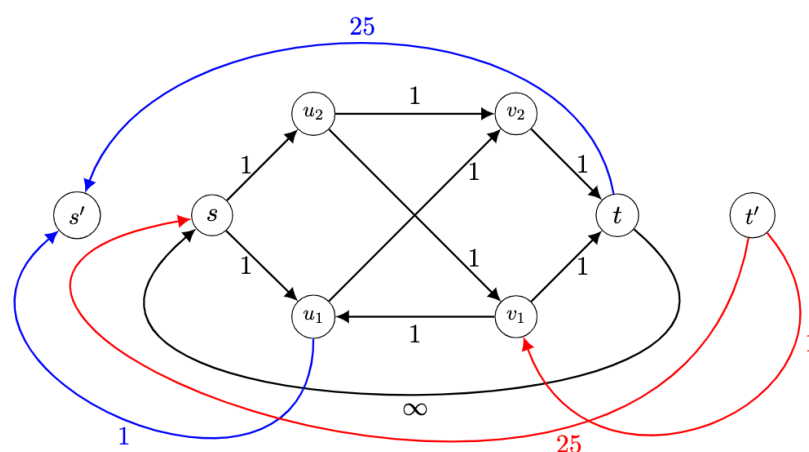
For å bestemme en tillatt strøm  $\tilde{x}$  i Figur 4.8 utvider vi nettverket med nodene  $s'$  og  $t'$  og legger til kantene  $(s', t)$  med kapasitet 25,  $(s', u_1)$  med kapasitet 1,  $(s, t')$  med kapasitet 25 og  $(v_1, t')$  med kapasitet 1. Det resulterer i nettverket vist i Figur 4.9.



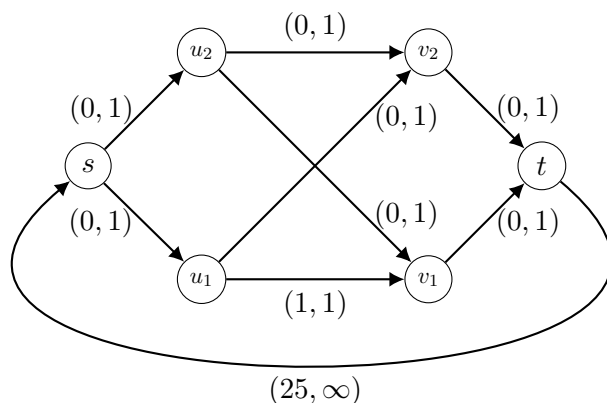
Figur 4.9: Nettverket i Figur 4.8 med to ekstra noder  $s'$  og  $t'$  og ekstra kanter  $(s', t)$ ,  $(s', u_1)$ ,  $(s, t')$  og  $(v_1, t')$ .

Videre bestemmer vi en maksimum  $s't'$ -strøm i dette nettverket ved å bruke Ford-Fulkerson Algoritmen:

- La  $\tilde{x} = 0$ . Residualnettverket til  $\tilde{x}$  er da det samme som det innledende nettverket i Figur 4.9. Vi identifiserer veien  $s' - t - s - t'$  som en tilføyende vei fra  $s'$  til  $t'$ : langs kantene  $(s', t)$ ,  $(t, s)$  og  $(s, t')$  kan  $\tilde{x}$  økes med 25. Videre identifiserer vi veien  $s' - u_1 - v_1 - t'$  som en tilføyende vei fra  $s'$  til  $t'$ : langs kantene  $(s', u_1)$ ,  $(u_1, v_1)$  og  $(v_1, t')$  kan  $\tilde{x}$  økes med 1. Residualnettverket til  $\tilde{x}$  etter disse økningene er vist i Figur 4.10.
- Vi ser at det ikke er noen gjenværende tilføyende vei i Figur 4.10, og konkluderer med at  $\tilde{x}$  er en en maksimum  $s't'$ -strøm. Fordi  $\tilde{x}$  metter alle kanter fra  $s'$  og alle kanter til  $t'$ , følger det fra Teorem 3.4.1 strømmen er tillatt i nettverket uten nodene  $s'$  og  $t'$ . Den tillatte strømmen i nettverket uten nodene  $s'$  og  $t'$  er vist i Figur 4.11.

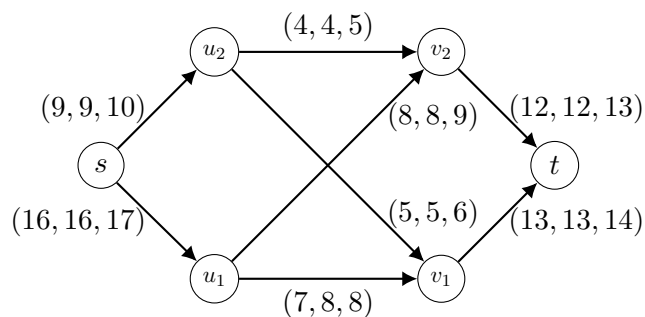


Figur 4.10: Residualnettverket til  $\tilde{x}$  etter å ha økt strømmen fra 0 til 25 langs kantene  $(s', t)$ ,  $(t, s)$  og  $(s, t')$  og fra 0 til 1 langs kantene  $(s', u_1)$ ,  $(u_1, v_1)$  og  $(v_1, t')$ .



Figur 4.11: En tillatt strøm  $\tilde{x}$  i nettverket i Figur 4.8

Til slutt setter vi løsningen for  $\tilde{x}$  inn i (4.5) og fjerner kanten fra  $t$  til  $s$ . Det resulterer i nettverket vist i Figur 4.12a med en strøm  $x$  som oppfyller både de nedre og øvre kapasitetsbegrensningene. Matrisen som korresponderer med strømmen  $x$  er vist Figur 4.12b. △



(a) Nettverket fra Figur 4.7a med nedre kapasitet, tillatt strøm og øvre kapasitet,  $(l(e), x(e), u(e))$  tilordnet kant  $e$ .

$$\begin{array}{c}
 v_1 \quad v_2 \\
 u_1 \begin{bmatrix} 8 & 8 \end{bmatrix} 16 \\
 u_2 \begin{bmatrix} 5 & 4 \end{bmatrix} 9 \\
 13 \quad 12
 \end{array}$$

(b) Matrisen som korresponderer med strømmen  $x$ .

Figur 4.12



## Kapittel 5

# Transportmatriser med trappemønster

Vi skal i dette kapittelet presentere teori for når det eksisterer transportmatriser der matriseelementene skal oppfylle gitte begrensninger. Vi introduserer først det som kalles et *trappemønster* (engelsk: staircase pattern), og videre et teorem fra [Dah08] som karakteriserer når det eksisterer matriser med gitte linjesummer og et slikt mønster. Avslutningsvis skal vi finne en transportmatrise med et bestemt trappemønster ved å løse et tillatt strøm problem.

### 5.1 Trappematriser

Definisjonene og notasjonen som introduseres i dette delkapittel er hentet fra [Dah08].

*Mønsteret* (engelsk: pattern) til en ikke-negativ  $m \times n$  matrise  $A = [a_{ij}]$  betegnes med  $\mathcal{P}(A)$ , og er definert som mengden av posisjonene til matrisens positive elementer, dvs.  $\mathcal{P}(A) = \{(i, j) : a_{ij} \neq 0\}$ . Dersom en matrise  $B$  har elementer i samme posisjoner som noen av elementene i  $A$  sier vi at mønsteret til  $B$  er inneholdt i mønsteret til  $A$ . Vi betegner det med  $\mathcal{P}(B) \subseteq \mathcal{P}(A)$ .

La  $v_i$  og  $h_i$ , være heltall s.a.

$$\begin{aligned} 1 \leq v_1 \leq v_2 \leq \dots \leq v_m \quad \text{og} \quad h_1 \leq h_2 \leq \dots \leq h_m = n, \\ \text{der } v_i \leq h_i \quad \text{for } i = 1, \dots, m \quad \text{og} \quad v_i \leq h_{i-1} + 1 \quad \text{for } i = 2, \dots, m. \end{aligned} \quad (5.1)$$

Definer  $m \times n$  matrisen  $W = [w_{ij}]$  som  $w_{ij} = 1$  hvis  $v_i \leq j \leq h_i$  og  $w_{ij} = 0$  ellers. Vi kaller  $W$  en trappematrise og mønsteret  $\mathcal{P}(W)$  et trappemønster.

For en trappematrise  $W$  definerer vi mengdene

$$\begin{aligned} C_W^1 &= \{(p, q) : w_{pq} = 1, w_{p,q+1} = 0, w_{p+1,q+1} = 1\}, \\ C_W^2 &= \{(p, q) : w_{pq} = 1, w_{p+1,q} = 0, w_{p+1,q+1} = 1\}, \end{aligned}$$

og kaller  $(p, q) \in C_W^1 \cup C_W^2$  en kritisk posisjon.

La  $R = (r_1, r_2, \dots, r_m)$  og  $S = (s_1, s_2, \dots, s_n)$  være ikke-negative heltallsvektorer med elementer slik at

$$\sum_{i=1}^m r_i = \sum_{j=1}^n s_j.$$

Vi betegner  $\mathcal{N}(R, S)$  som mengden transportmatriser med radsumvektor  $R$  og kolonnesumvektor  $S$ . Videre betegner vi  $\mathcal{N}_W(R, S)$  som mengden matriser i  $\mathcal{N}(R, S)$  som er slik at matrisens positive elementer har samme posisjon som enerne i  $W$ , dvs.

$$\mathcal{N}_W(R, S) = \{A \in \mathcal{N}(R, S) : \mathcal{P}(A) = \mathcal{P}(W)\}.$$

Tilsvarende betegner vi  $\mathcal{N}_{\subseteq W}(R, S)$  som mengden matriser i  $\mathcal{N}(R, S)$  som er slik at noen av matrisens positive elementer har samme posisjon som enerne i  $W$ , dvs.

$$\mathcal{N}_{\subseteq W}(R, S) = \{A \in \mathcal{N}(R, S) : \mathcal{P}(A) \subseteq \mathcal{P}(W)\}.$$

**Teorem 5.1.1.** (“Teorem 4” i [Dah08]). La  $W$  være en  $m \times n$  trappematrix. Da inneholder  $\mathcal{N}(R, S)$  en matrise  $A$  med  $\mathcal{P}(A) \subseteq \mathcal{P}(W)$  hvis og bare hvis  $R$  og  $S$  oppfyller

$$\begin{aligned} \sum_{i=1}^p r_i &\leq \sum_{j=1}^q s_j && ((p, q) \in C_W^1), \text{ og} \\ \sum_{i=1}^p r_i &\geq \sum_{j=1}^q s_j && ((p, q) \in C_W^2). \end{aligned} \tag{5.2}$$

For bevis, se [Dah08].

## 5.2 Strøm i nettverk og transportmatriser med trappemønstre

Hvis Teorem 5.1.1 gir at det eksisterer en matrise  $A \in \mathcal{N}_{\subseteq W}(R, S)$ , kan vi finne en slik ved å løse et tillatt strøm problem. Vi konstruerer først matrisens korresponderende nettverk som beskrevet i delkapittel 4.2. For å sikre at matrisen får riktig mønster tilordner vi kant  $(u_i, v_j)$  kapasitet 0 hvis  $w_{ij} = 0$  og kapasitet  $\min(r_i, s_j)$  ellers. Deretter bestemmer vi en tillatt strøm i dette nettverket.

Vi illustrerer metoden gjennom et eksempel.

**Eksempel 5.2.1.** Vi vil, dersom det eksisterer, finne en matrise  $A \in \mathcal{N}_{\subseteq W}(R, S)$ , der  $R = (3, 5, 7, 9)$  og  $S = (2, 5, 4, 8, 5)$  og

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

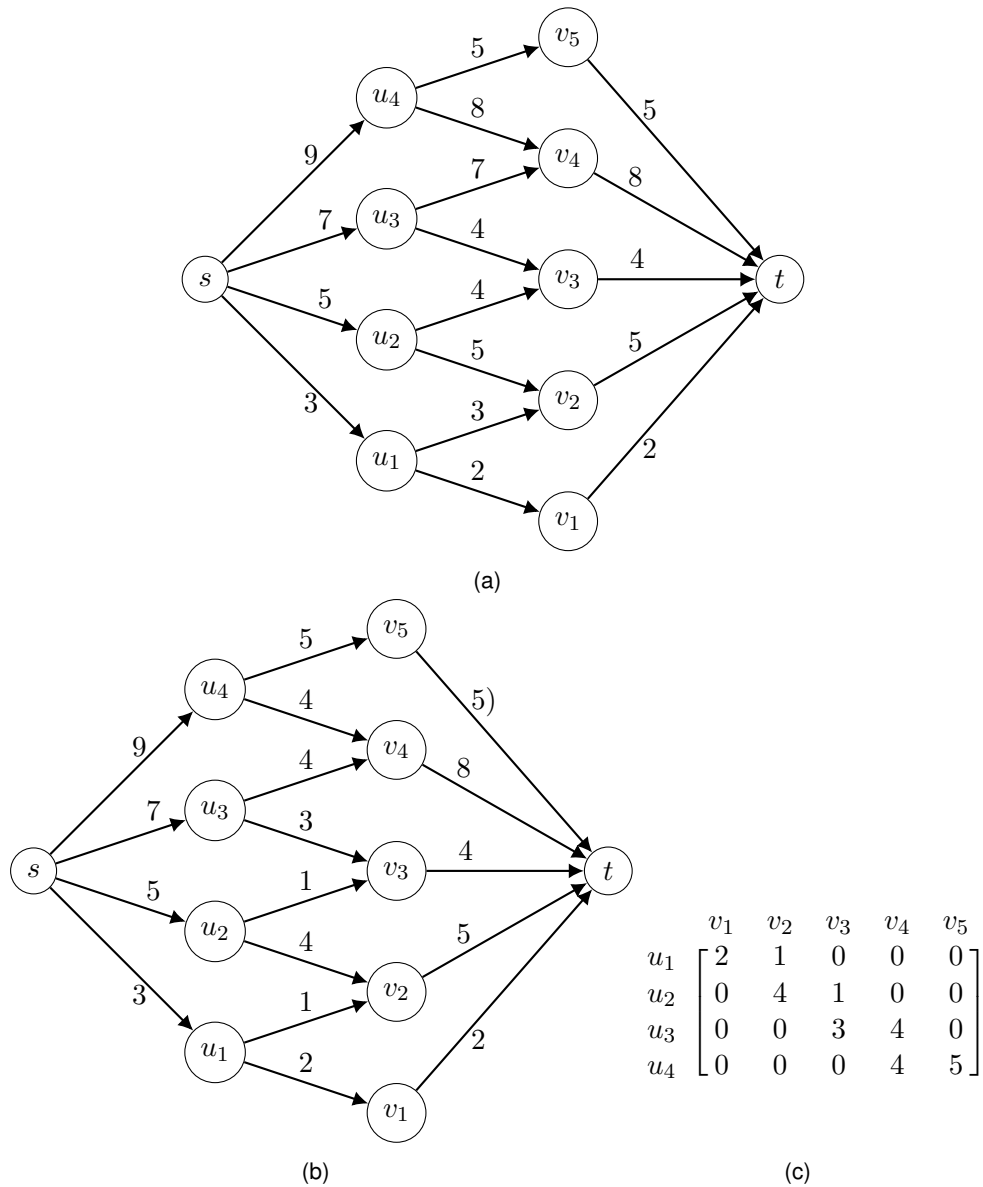
Vi bruker først Teorem 5.1.1 for å avgjøre om en slik matrise eksisterer. De kritiske posisjonene i  $W$  er  $C_W^1 = \{(1, 2), (2, 3), (3, 4)\}$  og  $C_W^2 = \{(1, 1), (2, 2), (3, 3)\}$ .

$$\begin{array}{ll} 3 < 2 + 5 & \checkmark & 3 > 2 & \checkmark \\ 3 + 5 < 2 + 5 + 4 & \checkmark & 3 + 5 > 2 + 5 & \checkmark \\ 3 + 5 + 7 < 2 + 5 + 4 + 8 & \checkmark & 3 + 5 + 7 > 2 + 5 + 4 & \checkmark \end{array}$$

Fordi  $R$  og  $S$  oppfyller betingelsene i Teorem 5.1.1, følger det at det eksisterer en matrise  $A \in \mathcal{N}_{\subseteq W}(R, S)$ . Vi finner én ved å løse et tillatt strøm problem. Det første vi gjør er å konstruere matrisens korresponderende nettverk der vi tilordner kant  $(u_i, v_j)$  kapasitet 0 hvis  $w_{ij} = 0$  og kapasitet  $\min(r_i, s_j)$  ellers. Videre utvider vi nettverket med to noder  $s$  og  $t$ , legger til en kant  $(s, u_i)$  for  $i = 1, 2, 3, 4$  med kapasitet lik henholdvis 3, 5, 7 og 9, og

## 5.2. Strøm i nettverk og transportmatriser med trappemønster

en kant  $(v_j, t)$  for  $j = 1, 2, 3, 4, 5$  med kapasitet lik henholdsvis 2, 5, 4, 8 og 5. Det resulterer i det transformerte nettverket vist i Figur 5.1a. Ved å bruke Ford Fulkerson algoritmen til å bestemme en maksimum  $st$ -strøm i dette nettverket, får vi  $st$ -strømmen  $x$  vist i Figur 5.1b. Fordi  $x$  metter alle kanter fra  $s$  og alle kanter til  $t$  følger det fra Teorem 3.4.1 at strømmen er tillatt i det originale nettverket. Matrisen som korresponderer med den tillatte strømmen er vist i Figur 5.1c.



**Figur 5.1:** (a) Det transformerte nettverket som korresponderer med en transportmatrise med radsumvektor  $R = (3, 5, 7, 9)$ , kolonnesumvektor  $S = (2, 5, 4, 8, 5)$  og trappemønster  $W$ . (b) Nettverket med en maksimum  $st$ -strøm  $x$  tilordnet hver kant. (c) Matrisen som korresponderer med  $x$ .

△





## Kapittel 6

# (0,1)-matriser med gitte linjesummer

Vi skal i dette kapitlet fortsette å diskutere transportmatriser, men begrense oss til matriser der elementene enten er 0 eller 1. Vi skal først løse et tillatt strøm problem for å avgjøre om det eksisterer, og i så fall finne en (0,1)-matrise med gitte linjesummer. Videre skal vi bruke Gale-Ryser teorem og algoritme for å gjøre det samme.

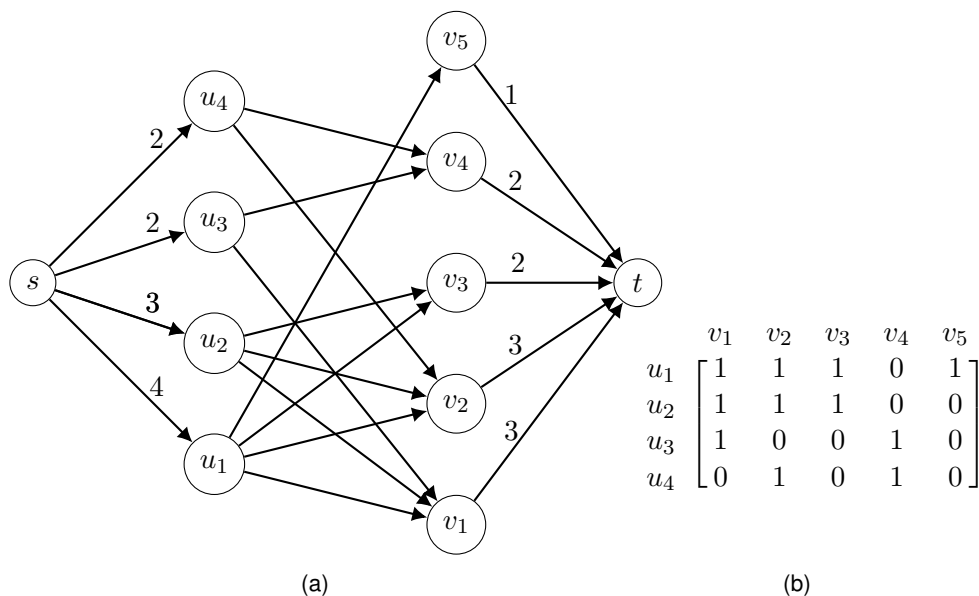
### 6.1 Strøm i nettverk og (0,1)-matriser med gitte linjesummer

Som beskrevet i delkapittel 4.2, kan vi avgjøre om det eksisterer, og i så fall finne en matrise inneholdt i  $\mathcal{N}(R, S)$  ved å løse et tillatt strøm problem. For å sørge for at matriseelementene som korresponderer med strømmen enten er 0 eller 1, tilordner vi hver kant i nettverket kapasitet 1. Videre bruker vi Ford-Fulkerson algoritmen for å bestemme en maksimum  $st$ -strøm  $x$  i det transformerte nettverket. Fordi alle kapasitetene langs kantene i det transformerte nettverket er heltall, vet vi fra beviset av Heltallsteoremet (Teorem 3.2.2) at  $x$  er heltallig, som vil si lik 0 eller 1 langs kantene i det originale nettverket. Dersom strømmen er tillatt i det originale nettverket, korresponderer strømmen med en (0,1)-matrise inneholdt i  $\mathcal{N}(R, S)$ .

**Eksempel 6.1.1.** La  $R = (4, 3, 2, 2)$  og  $S = (3, 3, 2, 2, 1)$ . Vi vil, dersom det eksisterer, finne en  $4 \times 5$  (0,1)-matrise med radsumvektor  $R$  og kolonnesumvektor  $S$ .

Vi starter med å konstruere matrisens korresponderende nettverk, der vi tilordner hver kant kapasitet 1. Videre utvider vi nettverket med to noder  $s$  og  $t$ , legger til en kant  $(s, u_i)$  for  $i = 1, 2, 3, 4$  med kapasitet lik henholdvis 4, 3, 2 og 2, og en kant  $(v_j, t)$  for  $j = 1, 2, 3, 4, 5$  med kapasitet lik henholdvis 3, 3, 2, 2 og 1. Ved å bruke Ford-Fulkerson algoritmen for å bestemme en maksimum  $st$ -strøm i det transformerte nettverket, får vi  $st$ -strømmen  $x$  vist i Figur 6.1a. Fordi  $x$  metter alle kanter fra node  $s$  og alle kanter til node  $t$  følger det fra Teorem 3.4.1 at  $x$  er tillatt i det originale nettverket. Matrisen som korresponderer med den tillatte strømmen er vist i Figur 6.1b.

△



Figur 6.1: a) En maksimum  $st$ -strøm  $x$  i det transformerte nettverket til en (0,1)-matrise med radsumvektor  $R = (4, 3, 2, 2)$  og kolonnesumvektor  $S = (3, 3, 2, 2, 1)$ . For hver kant  $(u_i, v_j)$  som er tegnet inn så er  $x(u_i, v_j) = 1$ . (b) Matrisen som korresponderer med  $x$ .

## 6.2 Gale-Ryser teorem og algoritme

Definisjonene og notasjonen som introduseres i dette delkapittelet er hentet fra [Bru06]. La  $x = (x_1, x_2, \dots, x_n)$  og  $y = (y_1, y_2, \dots, y_m)$  være to reelle ikke-økende vektorer, og la  $p = \max\{n, m\}$ . Vi sier at  $x$  er *majorisert* av  $y$ , betegnet med  $x \preceq y$ , hvis

$$\sum_{j=1}^k x_j \leq \sum_{j=1}^k y_j \quad (1 \leq k \leq p-1),$$

$$\sum_{j=1}^p x_j = \sum_{j=1}^p y_j,$$

der  $x_i = 0 \quad \forall \quad i > n$  og  $y_i = 0 \quad \forall \quad i > m$ .

La  $R = (r_1, r_2, \dots, r_m)$  være en ikke-økende, ikke-negativ heltallsvektor. Den *konjugerte* av  $R$  som vektoren  $R^* = \{r_1^*, r_2^*, \dots, r_m^*\}$ , der

$$r_k^* = |\{i : r_i \geq k\}| \quad (k \leq r_1).$$

Videre betegner vi  $\mathcal{A}(R, S)$  som mengden av alle (0,1)-matriser av størrelse  $m \times n$  med radsumvektor  $R = (r_1, \dots, r_m)$  og kolonnesumvektor  $S = (s_1, \dots, s_n)$ .

**Teorem 6.2.1.** (Gale-Ryser teorem, [Bru06, s. 27]). La  $R = (r_1, r_2, \dots, r_m)$  og  $S = (s_1, s_2, \dots, s_n)$  være to ikke-økende, ikke-negative heltallsvektorer. Da eksisterer det en matrise  $A \in \mathcal{A}(R, S)$  hvis og bare hvis

$$S \preceq R^*.$$

For bevis, se for eksempel [Kra96].

Vi illustrerer hvordan vi kan bruke Gale-Ryser teorem for å avgjøre om det eksisterer en  $(0,1)$ -matrise med gitte linjesummer gjennom et eksempel.

**Eksempel 6.2.2.** La  $R = (4, 3, 2, 2)$  og  $S = (3, 3, 2, 2, 1)$ .

Da er

$$\begin{aligned} r_1^* &= |\{i : r_i \geq 1\}| = 4 & r_3^* &= |\{i : r_i \geq 3\}| = 2 \\ r_2^* &= |\{i : r_i \geq 2\}| = 4 & r_4^* &= |\{i : r_i \geq 4\}| = 1 \end{aligned}$$

som gir  $R^* = (4, 4, 2, 1)$ . Videre er

$$\begin{aligned} 3 &< 4 \quad \checkmark \\ 3 + 3 &< 4 + 4 \quad \checkmark \\ 3 + 3 + 2 &< 4 + 4 + 2 \quad \checkmark \\ 3 + 3 + 2 + 2 &< 4 + 4 + 2 + 1 \quad \checkmark \\ 3 + 3 + 2 + 2 + 1 &= 4 + 4 + 2 + 1 + 0 \quad \checkmark \end{aligned}$$

som impliserer

$$S \preceq R^*.$$

Fra Teorem 6.2.1 følger det derfor at det finnes en  $4 \times 5$   $(0,1)$ -matrise med radsumvektor  $R = (4, 3, 2, 2)$  og kolonnesumvektor  $S = (3, 3, 2, 2, 1)$ .  $\triangle$

I tilfeller der Teorem 6.2.1 gir at det eksisterer en  $(0,1)$ -matrise med gitte linjesummer, kan vi i stedet for å løse et tillatt strøm problem, anvende Gale-Ryser algoritme for å finne én. Beskrivelsen av algoritmen er hentet fra [Bru06, s. 46].

### Gale-Ryser Algoritme

La  $R = (r_1, r_2, \dots, r_m)$  og  $S = (s_1, s_2, \dots, s_n)$  være to ikke-økende, ikke-negative heltallsvektorer.

1. Konstruer  $(0, 1)$ -matrisen av størrelse  $m \times n$ , der rad  $i$  består av  $r_i$  etterfølgende enere, etterfulgt av  $n - r_i$  nullere. Kall denne matrisen  $A(R, n)$ .
2. Konstruer en tom  $m \times 0$  matrise. Kall denne matrisen  $\tilde{A}$ .
3. For  $k = n, n - 1, \dots, 1$ :  
Flytt eneren som står lengst til høyre i de  $s_k$  radene i  $A(R, n)$  med størst sum, til kolonne  $k$ . Hvis to rader har lik sum: prioriter raden med størst indeks. Flytt videre kolonne  $k$  fra matrise  $A(R, n)$  til første kolonne i matrise  $\tilde{A}$ .

Etter  $n$  iterasjoner av punkt 3 blir  $\tilde{A}$  en  $m \times n$   $(0,1)$ -matrise med radsumvektor  $R$  og kolonnesumvektor  $S$ . Vi illustrerer algoritmen gjennom et eksempel.

**Eksempel 6.2.3.** Fra Eksempel 6.2.2 vet vi at det finnes en  $4 \times 5$   $(0,1)$ -matrise med radsumvektor  $R = (4, 3, 2, 2)$  og kolonnesumvektor  $S = (3, 3, 2, 2, 1)$ . Vi bruker Gale-Ryser Algoritme til å finne en slik matrise.

Vi konstruerer først  $A(R, 5)$  og en tom  $4 \times 0$  matrise,  $\tilde{A}$ . Videre gjør vi 5 iterasjoner av algoritmens tredje punkt:

$$\begin{array}{c} \left[ \begin{array}{ccccc} 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{array} \right] \xrightarrow{1.} \left[ \begin{array}{cccc|c} 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{array} \right] \xrightarrow{2.} \left[ \begin{array}{ccc|cc} 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{array} \right] \xrightarrow{3.} \\ \left[ \begin{array}{cc|ccc} 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \end{array} \right] \xrightarrow{4.} \left[ \begin{array}{c|cccc} 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{array} \right] \xrightarrow{5.} \left[ \begin{array}{cccc|c} 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{array} \right] \end{array}$$

1.  $k = 5$  og  $s_5 = 1$ . Fordi første rad har størst sum, flyttes eneren lengst til høyre i rad 1 til kolonne 5. Deretter flyttes kolonne 5 til første kolonne i  $\tilde{A}$ .
2.  $k = 4$  og  $s_4 = 2$ . Fordi de to første radene har størst sum, flyttes eneren lengst til høyre i rad 1 og rad 2 til kolonne 4. Deretter flyttes kolonne 4 til første kolonne i  $\tilde{A}$ .
3.  $k = 3$  og  $s_3 = 2$ . Fordi alle radene har lik sum, flyttes eneren lengst til høyre i rad 3 og 4 til kolonne 3. Deretter flyttes kolonne 3 til første kolonne i  $\tilde{A}$ .
4.  $k = 2$  og  $s_2 = 3$ . Fordi de to første radene har størst sum, flyttes eneren lengst til høyre i rad 1 og 2 til kolonne 2 (fordi disse enerne allerede står i kolonne 2, skjer det i prinsippet ingen endring.) Videre flyttes eneren lengst til høyre i rad 4 til kolonne 2. Deretter flyttes kolonne 2 til første kolonne i  $\tilde{A}$ .
5.  $k = 1$  og  $s_1 = 3$ . Fordi de tre første radene har størst sum, flyttes eneren lengst til høyre i rad 1, 2 og 3 til kolonne 1 (fordi disse enerne allerede står i kolonne 1, skjer det i prinsippet ingen endring). Deretter flyttes kolonne 1 til første kolonne i  $\tilde{A}$ .

Algoritmen resulterer i matrisen

$$\tilde{A} = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix}.$$

△

**Observasjon 6.2.4.** Matrisen  $A(R, n)$  (definert i punkt 1 av Gale-Ryser Algoritme) har  $r_k$  enere i rad  $k$ . Videre inneholder kolonne  $k$  én ener per rad med radsum større eller lik  $k$ , som vil si  $r_k^*$  enere. Dermed har  $A(R, n)$  radsumvektor  $R$  og kolonnesumvektor  $R^*$ . Fordi summen av rad- og kolonnesummene i  $A(R, n)$  er like, følger det at summen av elementene i  $R$  er lik summen av elementene  $R^*$ .

## Del II

# Presentasjon og diskusjon av egne resultater



## Kapittel 7

# Eksistens av transportmatriser med bestemte strukturer

Vi skal i dette kapitlet diskutere når det eksisterer diagonale, nedre triangulære, øvre triangulære og tridiagonale transportmatriser. Dette er alle kvadratiske matriser med nullere plassert i posisjoner til side for diagonalen. Mer presist:

- I en *diagonal matrise* er alle elementer utenom diagonalen lik null.
- I en *nedre triangulær matrise* er alle elementer over diagonalen lik null.
- I en *øvre triangulær matrise* er alle elementer under diagonalen lik null.
- I en *tridiagonal matrise* er alle elementer nedenfor den nedre diagonalen og ovenfor den øvre diagonalen lik null.

I første delkapittel løser vi ett tillatt strøm problem for å avgjøre om det eksisterer en diagonal, øvre triangulær, nedre triangulær og/eller tridiagonal transportmatrise med gitte linjesummer. I de tre neste delkapitlene undersøker vi om vi kan si noe generelt om hvilke betingelser to vektorer  $R$  og  $S$  må oppfylle for at det skal eksistere en nedre triangulær, øvre triangulær eller tridiagonal transportmatrise med linjesumvektorer  $R$  og  $S$ . I siste delkapittel presenteres en MATLAB-kode [MAT22] som sammenfatter kapitlet.

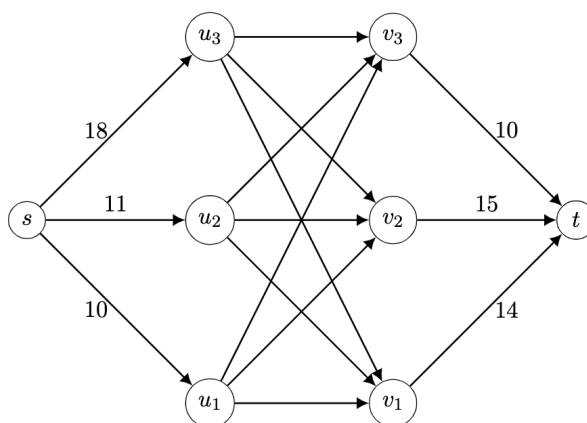
### 7.1 Strøm i nettverk og transportmatriser med bestemte strukturer

For matrisenetyperne beskrevet over merker vi oss følgende:

- Mønsteret til en diagonal matrise er inneholdt i mønsteret til en matrise  $D$  med mønster  $\mathcal{P}(D) = \{(i, j) : i = j\}$ .
- Mønsteret til en nedre triangulær matrise er inneholdt i mønsteret til en matrise  $N$  med mønster  $\mathcal{P}(N) = \{(i, j) : i \geq j\}$ .
- Mønsteret til en øvre triangulær matrise er inneholdt i mønsteret til en matrise  $O$  med mønster  $\mathcal{P}(O) = \{(i, j) : i \leq j\}$ .
- Mønsteret til en tridiagonal matrise er inneholdt i mønsteret til en matrise  $T$  med mønster  $\mathcal{P}(T) = \{(1, 1), (1, 2), \{(i, i - 1), (i, i), (i + 1) : i = 2, \dots, n - 1\}, (n, n - 1), (n, n)\}$ .

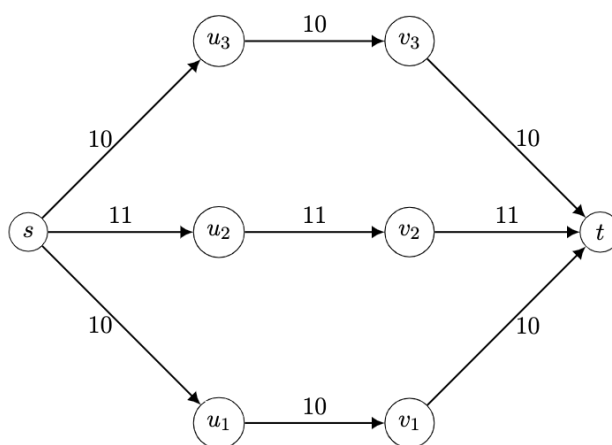
For avgjøre om det eksisterer, og i så fall bestemme en diagonal, nedre triangulær, øvre triangulær og/eller tridiagonal transportmatrise, tilordner vi kantene som korresponderer til posisjonene i henholdvis  $\mathcal{P}(D)$ ,  $\mathcal{P}(N)$ ,  $\mathcal{P}(O)$  og  $\mathcal{P}(T)$  øvre kapasitet 0.

**Eksempel 7.1.1.** La  $R = (10, 11, 18)$  og  $S = (14, 15, 10)$ . Vi konstruerer først det korresponderende nettverket til en  $3 \times 3$  transportmatrise med radsumvektor  $R$  og kolonnesumvektor  $S$ . Videre utvider vi nettverket med to noder  $s$  og  $t$ , legger til en kant  $(s, u_i)$  for  $i = 1, 2, 3$  med kapasitet lik henholdvis 10, 11 og 18, og en kant  $(v_j, t)$  for  $j = 1, 2, 3$  med kapasitet lik henholdvis 14, 15 og 10. Det gir det transformerte nettverket vist i Figur 7.1.



Figur 7.1: Det transformerte nettverket til en  $3 \times 3$  transportmatrise med radsumvektor  $R = (10, 11, 18)$  og kolonnesumvektor  $S = (14, 15, 10)$ .

1. For å se om det finnes en diagonal transportmatrise med linjesummer  $R$  og  $S$ , tilordner vi kant  $(u_i, v_j)$  i nettverket i Figur 7.1 kapasitet 0 for alle  $i \neq j$ . Ved å bruke Ford-Fulkerson algoritmen til å bestemme en maksimum  $st$ -strøm i dette nettverket, får vi  $st$ -strømmen  $x$  vist i Figur 7.2. Fordi  $x$  ikke metter kantene  $(s, u_3)$ ,  $(v_2, t)$  og  $(v_1, t)$ , er den ikke tillatt i det originale nettverket. Vi konkluderer derfor med at det ikke finnes en diagonal matrise med de gitte linjesummene.

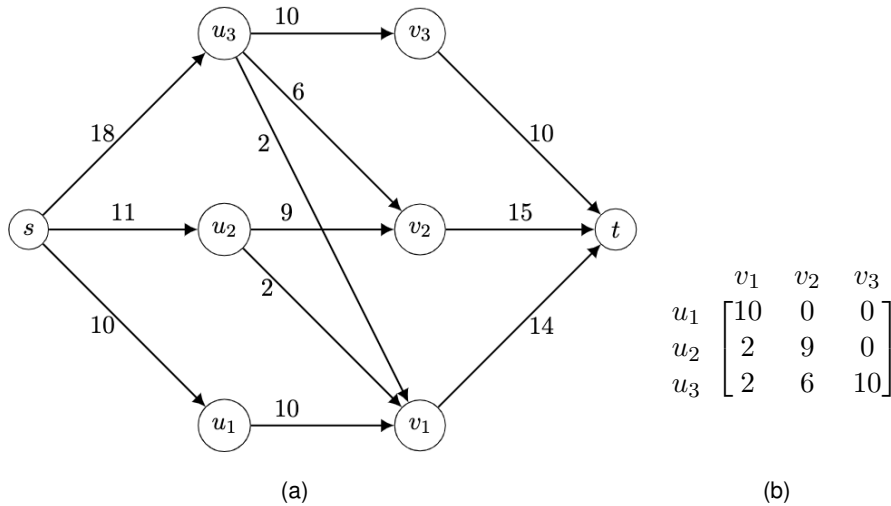


Figur 7.2: En maksimum  $st$ -strøm i nettverket i Figur 7.1 når kant  $(u_i, v_j)$  er tilordnet kapasitet 0 for alle  $i \neq j$ .



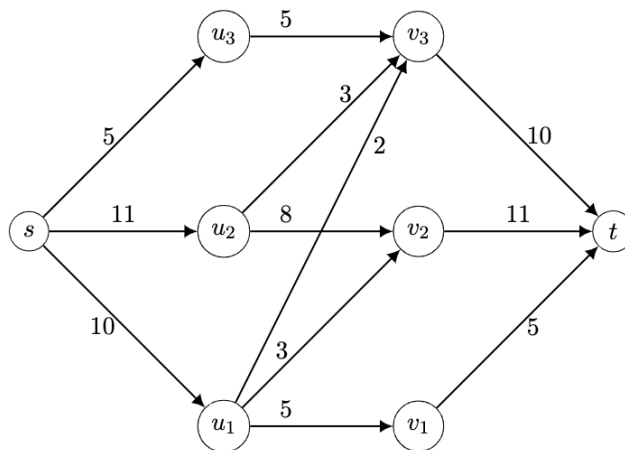
7.1. Strøm i nettverk og transportmatriser med bestemte strukturer

2. For å se om det finnes en nedre triangulær transportmatrise med linjesummer  $R$  og  $S$ , tilordner vi kantene  $(u_1, v_2)$ ,  $(u_1, v_3)$  og  $(u_2, v_3)$  i nettverket i Figur 7.1 kapasitet 0. Ved å bruke Ford-Fulkerson algoritmen til å bestemme en maksimum  $st$ -strøm i dette nettverket får vi  $st$ -strømmen  $x$  vist i Figur 7.3a. Fordi  $x$  metter alle kanter fra  $s$  og alle kanter til  $t$ , er den tillatt i det originale nettverket. Den nedre triangulære transportmatrisen som korresponderer med  $x$ , er vist i Figur 7.3b.



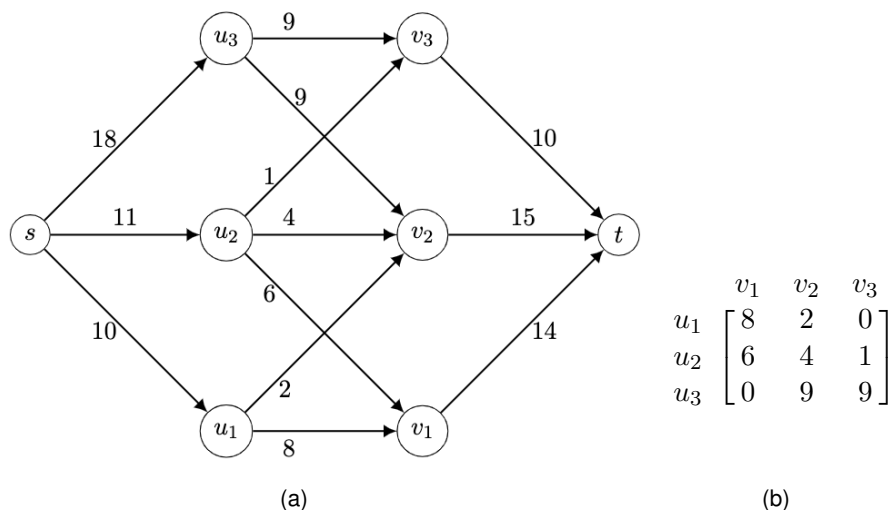
Figur 7.3: En maksimum  $st$ -strøm  $x$  i nettverket i Figur 7.1 når kantene  $(u_1, v_2)$ ,  $(u_1, v_3)$  og  $(u_2, v_3)$  er tilordnet kapasitet 0. b) Matrisen som korresponderer med  $x$ .

3. For å se om det finnes en øvre triangulær transportmatrise med linjesummer  $R$  og  $S$ , tilordner vi kantene  $(u_2, v_1)$ ,  $(u_3, v_1)$  og  $(u_3, v_2)$  i nettverket i Figur 7.1 kapasitet 0. Ved å bruke Ford-Fulkerson algoritmen til å bestemme en maksimum  $st$ -strøm i dette nettverket får vi  $st$ -strømmen  $x$  vist i Figur 7.4. Fordi  $x$  ikke metter kantene  $(s, u_3)$ ,  $(v_1, t)$  og  $(v_2, t)$ , er den ikke tillatt i det originale nettverket. Vi konkluderer derfor med at det ikke finnes en øvre triangulær med de gitte linjesumvektorene.



Figur 7.4: En maksimum  $st$ -strøm i nettverket i Figur 7.1 når kantene  $(u_2, v_1)$ ,  $(u_3, v_1)$  og  $(u_3, v_2)$  er tilordnet kapasitet 0.

4. For å se om det finnes en tridiagonal transportmatrise med linjesummer  $R$  og  $S$  tilordnet vi kantene  $(u_1, v_3), (u_3, v_1)$  i nettverket i Figur 7.1 kapasitet 0. Ved å bruke Ford-Fulkerson algoritmen til å bestemme en maksimum  $st$ -strøm i dette nettverket får vi  $st$ -strømmen  $x$  vist i Figur 7.5a. Fordi  $x$  metter alle kanter fra  $s$  og alle kanter til  $t$  er den tillatt i det originale nettverket. Den tridiagonale transportmatrisen som korresponderer med  $x$ , er vist Figur 7.5a.



Figur 7.5: En maksimum  $st$ -strøm i nettverket i Figur 7.1 når kantene  $(u_1, v_3), (u_3, v_1)$  er tilordnet kapasitet 0. b) Matrisen som korresponderer med  $x$ .

△

Uten å løse et tillatt strøm problem er det enkelt å se at to vektorer  $R = (r_1, r_2, \dots, r_n)$  og  $S = (s_1, s_2, \dots, s_n)$  må oppfylle  $r_i = s_i$  for  $i = 1, \dots, n$ , for at det skal eksistere en diagonal matrise med radsumvektor  $R$  og kolonnesumvektor  $S$ . Det er imidlertid ikke like opplagt hvilke betingelser  $R$  og  $S$  må oppfylle for at det skal eksistere en nedre triangulær, øvre triangulær eller tridiagonal matrise med linjesumvektorer  $R$  og  $S$ .

## 7.2 Eksistens av nedre triangulære transportmatriser

Vi skal i dette delkapittelet utforske om vi kan si noe generelt om hva som må kreves av to vektorer  $R = (r_1, r_2, \dots, r_n)$  og  $S = (s_1, s_2, \dots, s_n)$  for at det skal eksistere en nedre triangulær transportmatrise med radsumvektor  $R$  og kolonnesumvektor  $S$ .

Ved å sammenligne rad- og kolonnesummer innad i en nedre triangulær  $n \times n$  transportmatrise for  $n = 3, 4, 5$ , får vi sammenhengene vist i Figur 7.6. Generelt ser vi at de positive elementene i de  $k$  første radene er inneholdt i de  $k$  første kolonnene.

**Lemma 7.2.1.** For en ikke-negativ nedre triangulær matrise med radsumvektor  $R$  og kolonnesumvektor  $S$ , så er

$$\sum_{i=1}^k r_i \leq \sum_{j=1}^k s_j \quad \text{for } k = 1, 2, \dots, n-1. \quad (7.1)$$

## 7.2. Eksistens av nedre triangulære transportmatriser

*	0	0		$r_1$	
*	*	0		$r_2$	
*	*	*		$r_3$	
$s_1$	$s_2$	$s_3$			

$$r_1 \leq s_1$$

$$r_1 + r_2 \leq s_1 + s_2$$
  

*	0	0	0		$r_1$	
*	*	0	0		$r_2$	
*	*	*	0		$r_3$	
*	*	*	*		$r_4$	
$s_1$	$s_2$	$s_3$	$s_4$			

$$r_1 \leq s_1$$

$$r_1 + r_2 \leq s_1 + s_2$$

$$r_1 + r_2 + r_3 \leq s_1 + s_2 + s_3$$
  

*	0	0	0	0		$r_1$	
*	*	0	0	0		$r_2$	
*	*	*	0	0		$r_3$	
*	*	*	*	0		$r_4$	
*	*	*	*	*		$r_5$	
$s_1$	$s_2$	$s_3$	$s_4$	$s_5$			

$$r_1 \leq s_1$$

$$r_1 + r_2 \leq s_1 + s_2$$

$$r_1 + r_2 + r_3 \leq s_1 + s_2 + s_3$$

$$r_1 + r_2 + r_3 + r_4 \leq s_1 + s_2 + s_3 + s_4$$

Figur 7.6: Sammenhenger mellom rad- og kolonnesummer i en nedre triangulær  $3 \times 3$ ,  $4 \times 4$  og  $5 \times 5$  transportmatrise.

Med formål om å vise at Lemma 7.2.1 også gjelder den andre veien (dersom  $R$  og  $S$  oppfyller (7.1) så inneholder  $\mathcal{N}(R, S)$  en nedre triangulær matrise), undersøker vi hva slags matriser Nordvest-hjørne regelen produserer, gitt at  $R$  og  $S$  oppfyller (7.1).

For  $n = 3, 4, 5$ , viser det seg at hvis  $R$  og  $S$  oppfyller (7.1), så produserer Nordvest-hjørne regelen en nedre triangulær matrise (bevis gitt i Tillegg A.1). Det følger derfor at Lemma 7.2.1 i hvert fall gjelder den andre veien for  $n = 3, 4, 5$ . Fra beviset kommer det også frem at hvis  $R$  og  $S$  i tillegg til (7.1), oppfyller

$$\sum_{j=1}^k s_j \leq \sum_{i=1}^{k+1} r_i \quad \text{for } k = 1, \dots, n-2, \quad (7.2)$$

så produserer Nordvest-hjørne regelen en matrise  $A \in \mathcal{N}_W(R, S)$ , der  $W$  er trappematri- sen med  $v_1 = 1$ ,  $v_i = i - 1$  for  $i = 2, \dots, n$  og  $h_i = i$  for  $i = 1, \dots, n$ . Ved induksjon får vi at dette gjelder for alle verdier av  $n$  (induksjonsbevis gitt i Tillegg A.1). Fordi hver matrise  $A \in \mathcal{N}_W(R, S)$  er nedre triangulær, følger det at dersom  $R$  og  $S$  oppfyller (7.2) i tillegg til (7.1), så inneholder  $\mathcal{N}(R, S)$  en nedre triangulær matrise. For å vise at det gjelder uten å kreve at (7.2) er oppfylt, bruker vi Teorem 5.1.1.

La  $W$  være trappematri- sen vi får ved å la  $v_i = 1$  og  $h_i = i$  for  $i = 1, \dots, n$ . Ved å konstruere  $W$  ser vi at  $\mathcal{P}(W) = \{(i, j) : i \geq j\}$ , og at de kritiske posisjonene i  $W$  er  $C_W^1 = \{(1, 1), (2, 2), \dots, (n-1, n-1)\}$  og  $C_W^2 = \emptyset$ . Fra Teorem 5.1.1 følger det derfor

at  $\mathcal{N}(R, S)$  inneholder en matrise  $A$  med  $\mathcal{P}(A) \subseteq \mathcal{P}(W)$  hvis og bare hvis  $R$  og  $S$  oppfyller

$$\sum_{i=1}^k r_i \leq \sum_{j=1}^k s_j \quad \text{for } k = 1, \dots, n-1.$$

Fordi hver matrise  $A \in \mathcal{N}_{\subseteq W}(R, S)$  er nedre triangulær, kan vi utvide Lemma 7.2.1 til å gjelde begge veier og vi får følgende teorem.

**Teorem 7.2.2.**  $\mathcal{N}(R, S)$  inneholder en nedre triangulær matrise hvis og bare hvis  $R$  og  $S$  oppfyller

$$\sum_{i=1}^k r_i \leq \sum_{j=1}^k s_j \quad \text{for } k = 1, 2, \dots, n-1. \quad (7.1)$$

### 7.3 Eksistens av øvre triangulære transportmatriser

For å undersøke hvilke betingelser to vektorer  $R = (r_1, r_2, \dots, r_n)$  og  $S = (s_1, s_2, \dots, s_n)$  må oppfylle for at det skal eksistere en øvre triangulær transportmatrise med radsumvektor  $R$  og kolonnesumvektor  $S$ , utnytter vi relasjonen mellom en nedre triangulær matrise inneholdt i  $\mathcal{N}(S, R)$  og en øvre triangulær matrise inneholdt i  $\mathcal{N}(R, S)$ .

Fra Teorem 7.2.2 følger det at  $\mathcal{N}(S, R)$  inneholder en nedre triangulær matrise hvis og bare hvis

$$\sum_{j=1}^k s_j \leq \sum_{i=1}^k r_i \quad \text{for } k = 1, 2, \dots, n-1.$$

Videre vet vi at den transponerte matrisen til en nedre triangulær matrise i  $\mathcal{N}(S, R)$ , er en øvre triangulær matrise inneholdt i  $\mathcal{N}(R, S)$ . Dermed følger det at hvis  $\mathcal{N}(S, R)$  inneholder en nedre triangulær matrise, så inneholder  $\mathcal{N}(R, S)$  en øvre triangulær matrise, og vi får følgende teorem.

**Teorem 7.3.1.**  $\mathcal{N}(R, S)$  inneholder en øvre triangulær matrise hvis og bare hvis  $R$  og  $S$  oppfyller

$$\sum_{j=1}^k s_j \leq \sum_{i=1}^k r_i \quad \text{for } k = 1, 2, \dots, n-1. \quad (7.3)$$

Videre er det interessant å undersøke hva slags matriser Nordvest-hjørne regelen produserer hvis  $R$  og  $S$  oppfyller (7.3). For  $n = 3, 4, 5$  viser det seg at Nordvest-hjørne regelen da produserer en øvre triangulær matrise (bevis gitt i Tillegg A.2). Fra beviset kommer det også frem at hvis  $R$  og  $S$  i tillegg til (7.3), oppfyller

$$\sum_{i=1}^k r_i \leq \sum_{j=1}^{k+1} s_j \quad \text{for } k = 1, 2, \dots, n-2, \quad (7.4)$$

så produserer Nordvest-hjørne regelen en matrise  $A \in \mathcal{N}_W(R, S)$ , der  $W$  er trappematriksen med  $v_i = i$  for  $i = 1, \dots, n$  og  $h_i = i + 1$  for  $i = 1, \dots, n-1$ ,  $h_n = n$ . Ved induksjon får vi at dette gjelder for alle verdier av  $n$  (induksjonsbevis gitt i Tillegg A.2). Fordi hver matrise  $A \in \mathcal{N}_W(R, S)$  er øvre triangulær vet vi derfor at Nordvest-hjørne regelen produserer en øvre triangulær matrise hvis  $R$  og  $S$  oppfyller (7.3) og (7.4).

## 7.4 Eksistens av tridiagonale transportmatriser

Vi skal i dette delkapittelet utforske om vi kan si noe generelt om hvilke betingelser to vektorer  $R = (r_1, r_2, \dots, r_n)$  og  $S = (s_1, s_2, \dots, s_n)$  må oppfylle for at det skal eksistere en tridiagonal transportmatrise med radsumvektor  $R$  og kolonnesumvektor  $S$ .

$$\begin{array}{ccc|c} * & * & 0 & r_1 \\ * & * & * & r_2 \\ 0 & * & * & r_3 \\ \hline s_1 & s_2 & s_3 & \end{array} \qquad \begin{array}{l} r_1 \leq s_1 + s_2 \\ s_1 \leq r_1 + r_2 \end{array}$$

$$\begin{array}{cccc|c} * & * & 0 & 0 & r_1 \\ * & * & * & 0 & r_2 \\ 0 & * & * & * & r_3 \\ 0 & 0 & * & * & r_4 \\ \hline s_1 & s_2 & s_3 & s_4 & \end{array} \qquad \begin{array}{l} r_1 \leq s_1 + s_2 \\ r_1 + r_2 \leq s_1 + s_2 + s_3 \\ s_1 \leq r_1 + r_2 \\ s_1 + s_2 \leq r_1 + r_2 + r_3 \end{array}$$

$$\begin{array}{ccccc|c} * & * & 0 & 0 & 0 & r_1 \\ * & * & * & 0 & 0 & r_2 \\ 0 & * & * & * & 0 & r_3 \\ 0 & 0 & * & * & * & r_4 \\ 0 & 0 & 0 & * & * & r_5 \\ \hline s_1 & s_2 & s_3 & s_4 & s_5 & \end{array} \qquad \begin{array}{l} r_1 \leq s_1 + s_2 \\ r_1 + r_2 \leq s_1 + s_2 + s_3 \\ r_1 + r_2 + r_3 \leq s_1 + s_2 + s_3 + s_4 \\ s_1 \leq r_1 + r_2 \\ s_1 + s_2 \leq r_1 + r_2 + r_3 \\ s_1 + s_2 + s_3 \leq r_1 + r_2 + r_3 + r_4 \end{array}$$

Figur 7.7: Sammenhenger mellom rad- og kolonnesummer i en tridiagonal  $3 \times 3$ ,  $4 \times 4$  og  $5 \times 5$  transportmatrise.

Ved å sammenligne rad- og kolonnesummer innad i en tridiagonal  $n \times n$  transportmatrise for  $n = 3, 4, 5$ , får vi sammenhengene vist i Figur 7.7. Generelt ser vi at de positive elementene i de  $k$  første radene er inneholdt i de  $k + 1$  første kolonnene. Videre er de positive elementene i de  $k$  første kolonnene inneholdt i de  $k + 1$  første radene.

**Lemma 7.4.1.** For en ikke-negativ tridiagonal matrise med radsumvektor  $R = (r_1, r_2, \dots, r_n)$  og kolonnesumvektor  $S = (s_1, s_2, \dots, s_n)$  så er

$$\sum_{i=1}^k r_i \leq \sum_{j=1}^{k+1} s_j \quad \text{for } k = 1, 2, \dots, n-2, \quad \text{og} \quad (7.5a)$$

$$\sum_{j=1}^k s_j \leq \sum_{i=1}^{k+1} r_i \quad \text{for } k = 1, 2, \dots, n-2. \quad (7.5b)$$

Med formål om å vise at Lemma 7.4.1 også gjelder den andre veien (dersom to vektorer  $R$  og  $S$  oppfyller (7.5), så inneholder  $\mathcal{N}(R, S)$  en tridiagonal matrise), undersøker vi

hva slags matriser Nordvest-hjørne regelen produserer hvis  $R$  og  $S$  oppfyller (7.5). For  $n = 3, 4, 5$  viser det seg at hvis  $R$  og  $S$  oppfyller (7.5), så produserer Nordvest-hjørne regelen en tridiagonal matrise (bevis gitt i Tillegg A.3). Det følger derfor at Lemma 7.4.1 i hvert fall gjelder den andre veien for  $n = 3, 4, 5$ .

Videre ser vi at (7.5b) er samme ulikheter som (7.2). Vi har derfor fra delkapittel 7.2 at dersom  $R$  og  $S$  oppfyller (7.1) i tillegg til (7.5b), så produserer Nordvest-hjørne regelen en matrise  $A \in \mathcal{N}_W(R, S)$ , der  $W$  er trappematriksen med  $v_1 = 1$ ,  $v_i = i - 1$  for  $i = 2, \dots, n$  og  $h_i = i$  for  $i = 1, \dots, n$ . I tillegg til å være nedre triangulær, er hver matrise  $A \in \mathcal{N}_W(R, S)$  tridiagonal.

Tilsvarende har vi at (7.5a) er samme ulikheter som (7.4). Vi vet derfor fra delkapittel 7.3 at hvis  $R$  og  $S$  oppfyller (7.3) i tillegg til (7.5a), så produserer Nordvest-hjørne regelen en matrise  $A \in \mathcal{N}_W(R, S)$ , der  $W$  er trappematriksen med  $v_i = i$  for  $i = 1, \dots, n$  og  $h_i = i + 1$  for  $i = 1, \dots, n - 1$ ,  $h_n = n$ . I tillegg til å være øvre triangulær, er hver matrise  $A \in \mathcal{N}_W(R, S)$  tridiagonal.

Fordi (7.1) og (7.4) er strengere ulikheter enn henholdsvis (7.5a) og (7.5b), kan vi oppsummere observasjonene over som følger: Hvis  $R$  og  $S$  enten oppfyller (7.1) eller (7.3) i tillegg til (7.5), så inneholder  $\mathcal{N}(R, S)$  en tridiagonal matrise. For å vise at det gjelder uten å kreve at (7.1) eller (7.3) er oppfylt, bruker vi Teorem 5.1.1.

La  $W$  være trappematriksen vi får ved å la  $v_1 = 1$ ,  $v_i = i - 1$  for  $i = 2, \dots, n$ ,  $h_i = i + 1$  for  $i = 1, \dots, n - 1$ , og  $h_n = n$ . Ved å konstruere  $W$  ser vi at  $\mathcal{P}(W) = \{(1, 1), (1, 2), \{(i, i - 1), (i, i), (i + 1): i = 2, \dots, n - 1\}, (n, n - 1), (n, n)\}$ , og at de kritiske posisjonene i  $W$  er  $C_W^1 = \{(1, 2), (2, 3), \dots, (n - 2, n - 1)\}$  og  $C_W^2 = \{(2, 1), (3, 2), \dots, (n - 1, n - 2)\}$ . Fra Teorem 5.1.1 følger det derfor at  $\mathcal{N}(R, S)$  inneholder en matrise  $A$  med  $\mathcal{P}(A) \subseteq \mathcal{P}(W)$  hvis og bare hvis  $R$  og  $S$  oppfyller

$$\begin{aligned} \sum_{i=1}^k r_i &\leq \sum_{j=1}^{k+1} s_j & \text{for } k = 1, 2, \dots, n - 2, \\ \sum_{j=1}^k s_j &\leq \sum_{i=1}^{k+1} r_i & \text{for } k = 1, 2, \dots, n - 2. \end{aligned}$$

Fordi hver matrise  $A \in \mathcal{N}_{\subseteq W}(R, S)$  er tridiagonal, kan vi utvide Lemma 7.4.1 til å gjelde begge veier og vi får følgende teorem.

**Teorem 7.4.2.**  $\mathcal{N}(R, S)$  inneholder en tridiagonal matrise hvis og bare hvis  $R$  og  $S$  oppfyller

$$\begin{aligned} \sum_{i=1}^k r_i &\leq \sum_{j=1}^{k+1} s_j & \text{for } k = 1, 2, \dots, n - 2, & \text{og} \\ \sum_{j=1}^k s_j &\leq \sum_{i=1}^{k+1} r_i & \text{for } k = 1, 2, \dots, n - 2. \end{aligned} \tag{7.5}$$

## 7.5 MATLAB-program for å konstruere nedre triangulære, øvre triangulære og tridiagonale transportmatriser

Basert på diskusjonen tidligere i kapittelet har jeg brukt MATLAB [MAT22] til å skrive en kode som avgjør om det eksisterer, og i så fall lager en ikke-negativ nedre triangulær, øvre triangulær eller tridiagonal matrise med gitte rad- og kolonnesummer. Videre presenteres koden i korte trekk, mens den i sin helhet med beskrivende kommentarer er gitt i Tillegg B.

Koden består av en hovedfunksjon (gitt i Tillegg B.4) som tar to vektorer  $R$  og  $S$  som argumenter. Funksjonen sjekker først om  $R$  og  $S$  har lik lengde, og deretter om summen av elementene i  $R$  er lik summen av elementene i  $S$ . Dersom begge tilfellene oppfylles, kaller funksjonen på egendefinerte funksjoner (gitt i Tillegg B.1, Tillegg B.2, Tillegg B.3) som sjekker om  $R$  og  $S$  oppfyller (7.1), (7.3) eller (7.5). Dersom ingen av resultatene oppfylles returnerer funksjonen en kommentar om at det ikke eksisterer noen nedre triangulær, øvre triangulær eller tridiagonal matrise med de gitte linjesummene. Dersom et av resultatene oppfylles, løser funksjonen et tillatt strøm problem definert på nettverket som korresponderer med matrisen med den bestemte strukturen og linjesumvektorer  $R$  og  $S$ . Funksjonen returnerer til slutt en figur av det transformerte nettverket med den beregnede maksimum  $st$ -strømmen, og den korresponderende matrisen. I nettverket som returneres betegnes node  $s$  med 1, node  $t$  med  $2n + 2$ , radnode  $i$  med  $1 + i$  og kolonnenode  $j$  med  $1 + n + j$ .

I koden anvendes de innebygde MATLAB funksjonene *length*, *sum*, *zeros*, *digraph*, *plot*, *maxflow*, *highlight*, *labeledge* og *camroll* [MAT22].

Vi bruker MATLAB-koden til å se om det eksisterer, og i så fall bestemme en ikke-negativ nedre triangulær, øvre triangulær eller tridiagonal matrise med to ulike par av linjesumvektorer  $R$  og  $S$ .

**Eksempel 7.5.1.** La  $R = (7, 11, 2, 5)$  og  $S = (8, 7, 2, 8)$ .

```
>> makematrix(R,S)
```

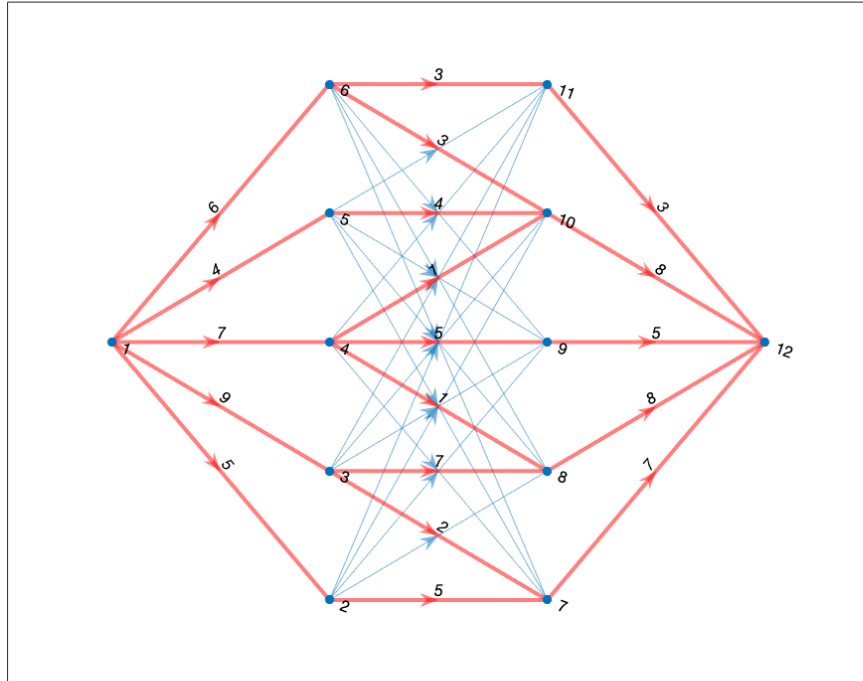
```
Det finnes ingen nedre triangulær, øvre triangulær eller  
tridiagonal ikke-negativ 4 x 4 matrise med de gitte linjesummene.
```

△

Kapittel 7. Eksistens av transportmatriser med bestemte strukturer

Eksempel 7.5.2. La  $R = (5, 9, 7, 4, 6)$  og  $S = (7, 8, 5, 8, 3)$ .

>> makematrix(R,S)



Figur 7.8: MATLAB-plot.

En tridiagonal 5 x 5 matrise med de gitte linjesummene er matrisen:

ans =

5	0	0	0	0
2	7	0	0	0
0	1	5	1	0
0	0	0	4	0
0	0	0	3	3

△



## Kapittel 8

# Eksistens av (0,1)-matriser med to, tre og fire rader og gitte linjesummer

Vi skal i dette kapitlet begrense oss til å se på (0,1)-matriser med to, tre og fire rader, og utforske hvilke betingelser to vektorer  $R$  og  $S$  da må oppfylle for at det skal eksistere en (0,1)-matrise med radsumvektor  $R$  og kolonnesumvektor  $S$ . Hensikten er å se om Gale-Ryser teorem (Teorem 6.2.1) kan forenkles når antall rader reduseres. Måten vi skal gjøre dette på er å fordele nullere og enere i  $n \times m$  matriser for  $m = 2, 3, 4$ , på en sånn måte at matrisene får radsumvektor  $R = (r_1, r_2, \dots, r_m)$  og kolonnesumvektor  $S = (s_1, s_2, \dots, s_n)$ . Fordi kolonner med kolonnesum  $m$  eller  $0$  er entydig bestemt, fjerner vi midlertidig disse kolonnene. Problemet reduseres dermed til å fordele nullere og enere i de resterende kolonnene.

I hele kapitlet antar vi at  $R = (r_1, r_2, \dots, r_m)$  og  $S = (s_1, s_2, \dots, s_n)$  er to ikke-økende, ikke-negative heltallsvektorer som oppfyller

$$\sum_{j=1}^n s_j = \sum_{i=1}^m r_i.$$

Vi definerer  $p$  og  $q$  til å være antall elementer i  $S$  som er lik henholdsvis  $m$  og  $0$ . Videre definerer vi  $n' = n - p - q$  og  $R' = (r'_1, r'_2, \dots, r'_m)$ , der  $r'_i = r_i - p$  for  $i = 1, 2, \dots, m$ . For tilfellene  $m = 3$  og  $m = 4$  definerer vi  $l$  til å være antall elementer i  $S$  som er lik  $2$ . For  $m = 4$  definerer vi i tillegg  $k$  til å være antall elementer i  $S$  som er lik  $3$ . Fordi den videre diskusjonen er avhengig av om  $m$  er lik  $2, 3$  eller  $4$ , tar vi for oss hvert tilfelle i ulike delkapitler.

### 8.1 Eksistens av (0,1)-matriser med to rader og gitte linjesummer

**Teorem 8.1.1.** *La  $R = (r_1, r_2)$  og  $S = (s_1, s_2, \dots, s_n)$  være to ikke-økende, ikke-negative heltallsvektorer med*

$$\sum_{j=1}^n s_j = r_1 + r_2.$$

Da eksisterer det en  $2 \times n$  (0,1)-matrise  $A = [a_{ij}]$  med

$$\sum_{j=1}^n a_{ij} = r_i \quad (i \leq 2)$$

$$\sum_{i=1}^2 a_{ij} = s_j \quad (j \leq n)$$

hvis og bare hvis

$$r_1 \leq n - q. \quad (8.1)$$

*Bevis.* Anta at  $A = [a_{ij}]$  er en  $2 \times n$  (0,1)-matrise med radsumvektor  $R$  og kolonnesumvektor  $S$ . Da er  $a_{1j} = 0$  for  $j > n - q$ , og fordi resten av elementene i første rad enten er lik 1 eller 0, er  $r_1 \leq n - q$ .

For å vise den andre veien, anta at  $R$  og  $S$  oppfyller (8.1). Konstruer nullmatrisen av størrelse  $2 \times n'$ . Erstatt nullene i de  $r'_1$  første kolonnene av rad 1 med enere. Fra (8.1) følger det at  $r'_1 \leq n'$ , som bekrefter at det er plass til  $r'_1$  enere i første rad. Erstatt deretter nullene i de  $n' - r_1$  siste kolonnene av rad 2 med enere. Fra (8.2) ser vi at antall nullere som erstattes med enere i rad 2 er lik  $r'_2$ .

$$r_1 + r_2 = \sum_{j=1}^n s_j = 2 \cdot p + 1 \cdot (n - p - q) + 0 \cdot q$$

$$r'_1 + p + r'_2 + p = 2p + n'$$

$$r'_2 = n' - r'_1. \quad (8.2)$$

Den konstruerte matrisen har dermed nøyaktig én ener i hver kolonne, og radsummer lik henholdsvis  $r'_1$  og  $r'_2$ . Ved å til slutt legge  $p$  kolonner til venstre med enere i begge rader og  $q$  kolonner til høyre med nullere i begge rader, får vi en  $2 \times n$  matrise med radsumvektor  $R$  og kolonnesumvektor  $S$ . ■

## 8.2 Eksistens av (0,1)-matriser med tre rader og gitte linjesummer

**Teorem 8.2.1.** La  $R = (r_1, r_2, r_3)$  og  $S = (s_1, s_2, \dots, s_n)$  være to ikke-økende, ikke-negative heltallsvektorer med

$$\sum_{j=1}^n s_j = r_1 + r_2 + r_3.$$

Da eksisterer det en  $3 \times n$  (0,1)-matrise  $A = [a_{ij}]$  med

$$\sum_{j=1}^n a_{ij} = r_i \quad (i \leq 3)$$

$$\sum_{i=1}^3 a_{ij} = s_j \quad (j \leq n)$$

hvis og bare hvis

$$r_1 \leq n - q, \quad (8.3a)$$

$$r_3 \geq p. \quad (8.3b)$$

*Bevis.* Anta at  $A = [a_{ij}]$  er en  $3 \times n$  (0,1)-matrise med radsumvektor  $R$  og kolonnesumvektor  $S$ . Da er  $a_{1j} = 0$  for  $j > n - q$ , og fordi resten av elementene i første rad enten er lik 1 eller 0, er  $r_1 \leq n - q$ . Videre er  $a_{3j} = 1$  for  $j \leq p$ , som gir  $r_3 \geq p$ .

For å vise den andre veien, anta at  $R$  og  $S$  oppfyller (8.3). Konstruer nullmatrisen av størrelse  $3 \times n'$  og erstatt nullene i de  $r'_1$  første kolonnene av rad 1 med enere. Fra (8.3a) følger det at  $r'_1 \leq n'$ , som bekrefter at det er plass til  $r'_1$  enere i første rad. Vi skal videre erstatte nullere med enere i rad 2 og 3 på en sånn måte at vi først fullfører kolonnene som skal ha kolonnesum 2, før kolonnene som skal ha kolonnesum 1. Fordi måten dette gjøres på avhenger av størrelsen på  $r'_1$  og  $r'_2$  i forhold til  $l$ , gis det bevis for de ulike tilfellene. For alle tilfellene trenger vi følgende sammenheng.

$$\begin{aligned} r_1 + r_2 + r_3 &= \sum_{j=1}^n s_j \\ r_1 + r_2 + r_3 &= 3 \cdot p + 2 \cdot l + 1 \cdot (n - p - l - q) + 0 \cdot q \\ r'_1 + p + r'_2 + p + r'_3 + p &= 3p + n' + l \\ r'_1 + r'_2 + r'_3 &= n' + l \\ r'_3 &= n' + l - r'_1 - r'_2. \end{aligned} \tag{8.4}$$

**Tilfelle 1:** Anta  $r'_1 \geq l$ .

1. Rad 2: Erstatt nullene fra og med kolonne 1 mot kolonne  $l$  med enere. Hvis  $r_2 > l$ : erstatt nullene fra og med kolonne  $r_1 + 1$  med enere. Med  $r'_3 \geq 0$  innsatt i (8.4) følger det at  $r'_2 \leq l + n' - r'_1$  som bekrefter at det er plass til  $r'_2$  enere i andre rad.
2. Rad 3:
  - (a) Hvis  $r_2 \geq l$ : erstatt nullene fra og med kolonne  $r_1 + (r_2 - l) + 1$  til og med kolonne  $n'$  med enere. Det innebærer å erstatte  $n' - r_1 - (r_2 - l) = n' + l - r_1 - r_2$  nullere med enere. Fra (8.4) ser vi at rad 3 fylles opp med nøyaktig  $r'_3$  enere.
  - (b) Hvis  $r_2 < l$ : erstatt nullene fra og med kolonne  $r_2 + 1$  til og med kolonne  $l$  med enere. Erstatt deretter nullene fra og med kolonne  $r_1 + 1$  til og med kolonne  $n'$  med enere. Dette innebærer å erstatte  $l - r_2 + n' - r_1 = n' + l - r_1 - r_2$  nullere med enere. Fra (8.4) ser vi at rad 3 fylles opp med nøyaktig  $r'_3$  enere.

**Tilfelle 2:** Anta  $r'_1 < l$ .

1. Rad 2: Erstatt nullene fra og med kolonne  $r'_1 + 1$  til og med kolonne  $l$  med enere. Fra  $r'_3 \leq r'_1 < l \leq n'$  får vi  $r'_3 < n'$ . Med  $r'_3 < n'$  innsatt i (8.4) følger det at  $r'_2 > l - r'_1$ . Erstatt derfor også nullene i de  $r'_2 - (l - r'_1)$  første kolonnene med enere.
2. Rad 3: Erstatt nullene fra og med kolonne  $r'_2 - (l - r'_1) + 1$  til og med kolonne  $l$  med enere. Erstatt videre nullene i de  $n' - l$  siste kolonnene med enere. Totalt innebærer det å erstatte  $l - (r'_1 + r'_2 - l) + n' - l = n' + l - r'_1 - r'_2$  nullere med enere. Fra (8.4) ser vi at rad 3 fylles opp med nøyaktig  $r'_3$  enere.

For alle tilfellene får vi en  $3 \times n'$  matrise der de  $l$  første kolonnene har kolonnesum lik 2 og resten har kolonnesum lik 1. Videre har matrisen radsummer lik henholdsvis  $r'_1, r'_2$  og  $r'_3$ . Ved å til slutt legge  $p$  kolonner til venstre med enere i alle rader og  $q$  kolonner til høyre med nullere i alle rader, får vi en  $3 \times n$  (0,1)-matrise med radsumvektor  $R$  og kolonnesumvektor  $S$ . ■

Vi illustrerer algoritmen beskrevet for “Tilfelle 2” gjennom et eksempel.

**Eksempel 8.2.2.** La  $R = (7, 7, 5)$  og  $S = (3, 2, 2, 2, 2, 2, 2, 2, 1, 1, 0)$ .

Vi sjekker først at  $R$  og  $S$  oppfyller antakelsene algoritmen er basert på, det vil si (8.3a) og (8.3b).

$$\begin{aligned} r_1 &= 7 < n - q = 11 - 1 = 10 \quad \checkmark \\ r_3 &= 5 > p = 1 \quad \checkmark. \end{aligned}$$

Fordi den første og den siste kolonnen er entydig bestemt, holder det å finne en  $3 \times 9$  matrise med radsumvektor  $R' = (6, 6, 4)$  og kolonnesumvektor  $S' = (2, 2, 2, 2, 2, 2, 2, 1, 1)$ .

1. Vi starter med å konstruere nullmatrisen av størrelse  $3 \times 9$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

2. Rad 1: Vi erstatter nullene i de 6 første kolonnene med enere.
3. Rad 2: Vi erstatter nullen i kolonne 7 med én ener. Deretter erstatter vi nullene i de  $6 - (7 - 6) = 5$  første kolonnene med enere.
4. Rad 3: Vi erstatter nullene fra og med kolonne  $5 + 1 = 6$  til og med kolonne 7 med enere, og nullene i de to siste kolonnene med enere.

Totalt resulterer endringene i rad 1, 2 og 3, i matrisen

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

Til slutt legger vi til én kolonne til venstre med enere i alle rader, og én kolonne til høyre med nullere i alle rader. Det gir matrisen

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}.$$

△

### 8.3 Eksistens av (0,1)-matriser med fire rader og gitte linjesummer

**Teorem 8.3.1.** La  $R = (r_1, r_2, r_3, r_4)$  og  $S = (s_1, s_2, \dots, s_n)$  være to ikke-økende, ikke-negative heltallsvektorer med

$$\sum_{j=1}^n s_j = r_1 + r_2 + r_3 + r_4.$$

Da eksisterer det en  $4 \times n$  (0,1)-matrise  $A = [a_{ij}]$  med

$$\begin{aligned} \sum_{j=1}^n a_{ij} &= r_i \quad (i \leq 4) \\ \sum_{i=1}^4 a_{ij} &= s_j \quad (j \leq n) \end{aligned}$$

### 8.3. Eksistens av (0,1)-matriser med fire rader og gitte linjesummer

hvis og bare hvis

$$r_1 \leq n - q, \quad (8.5a)$$

$$r_4 \geq p, \quad (8.5b)$$

$$r_1 + r_2 \leq n - q + p + k + l. \quad (8.5c)$$

*Bevis.* Anta at  $A = [a_{ij}]$  er en  $4 \times n$  (0,1)-matrise med radsumvektor  $R$  og kolonnesumvektor  $S$ . Da er  $a_{1j} = 0$  for  $j > n - q$ , og fordi resten av elementene i første rad enten er lik 1 eller 0, er  $r_1 \leq n - q$ . Videre er  $a_{4j} = 1$  for  $j \leq p$ , som gir  $r_4 \geq p$ . Til slutt har vi at de  $p$  første kolonnene har enere i alle rader, de  $k + l$  neste kolonnene kan ha enere i to rader per kolonne, de  $n - p - k - l - q$  neste kolonnene kan kun ha én ener i en rad per kolonne, mens de  $q$  siste kolonnene har nullere i alle rader. Dermed er

$$r_1 + r_2 \leq 2 \cdot p + 2 \cdot (k + l) + 1 \cdot (n - p - k - l - q) + 0 \cdot q = n - q + p + k + l.$$

For å vise den andre veien, anta at  $R$  og  $S$  oppfyller (8.5). Konstruer nullmatrisen av størrelse  $4 \times n'$  og erstatt nullene i de  $r'_1$  første kolonnene av rad 1 med enere. Fra (8.5a) følger det at  $r'_1 \leq n'$ , som bekrefter at det er plass til  $r'_1$  enere i første rad. Vi skal videre erstatte nullere med enere på en sånn måte at vi først fullfører kolonnene som skal ha kolonnesum lik 2 og 3, før kolonnene som skal ha kolonnesum lik 1. Fordi måten dette gjøres på avhenger av størrelsen på  $r'_1$ ,  $r'_2$  og  $r'_3$  i forhold til  $l$  og  $k$ , gis det bevis for de ulike tilfellene. For alle tilfellene trenger vi følgende sammenheng.

$$\begin{aligned} r_1 + r_2 + r_3 + r_4 &= \sum_{j=1}^n s_j \\ r_1 + r_2 + r_3 + r_4 &= 4 \cdot p + 3 \cdot k + 2 \cdot l + 1 \cdot (n - p - k - l - q) + 0 \cdot q \\ r'_1 + p + r'_2 + p + r'_3 + p + r'_4 + p &= 4p + n' + 2k + l \\ r'_1 + r'_2 + r'_3 + r'_4 &= n' + 2k + l \\ r'_4 &= n' + 2k + l - r'_1 - r'_2 - r'_3. \end{aligned} \quad (8.6)$$

**Tilfelle 1:** Anta  $r'_1, r'_2 \geq k + l$ .

1. Rad 2: Erstatt nullene i de  $k + l$  første kolonnene med enere. Hvis  $r'_2 > k + l$ : erstatt nullene fra og med kolonne  $r'_1 + 1$  med enere. Fra (8.5c) følger det at  $r'_2 \leq n' - r'_1 + k + l$  som bekrefter at det er plass til  $r'_2$  enere i andre rad.
2. Rad 3: Erstatt nullene fra og med kolonne 1 mot kolonne  $k$  med enere. Hvis  $r'_3 > k$ : erstatt nullene fra og med kolonne  $r'_1 + r'_2 - (k + l) + 1$  mot kolonne  $n'$  med enere. Med  $r'_4 \geq 0$  innsatt i (8.6) følger det at  $r'_3 \leq n' + 2k + l - r'_1 - r'_2$ , som gir at det er plass til  $r'_3$  enere i tredje rad.
3. Rad 4:
  - (a) Hvis  $r'_3 \geq k$ : Erstatt nullene fra og med kolonne  $r'_1 + r'_2 - (k + l) + (r'_3 - k) + 1$  til og med kolonne  $n'$  med enere. Dette innebærer å erstatte  $n' - (r'_1 + r'_2 - (k + l) + (r'_3 - k)) = n' + 2k + l - r'_1 - r'_2 - r'_3$  nullere med enere. Fra (8.6) ser vi at rad 4 fylles opp med nøyaktig  $r'_4$  enere.
  - (b) Hvis  $r'_3 < k$ : Erstatt nullene fra og med kolonne  $r'_3 + 1$  til og med kolonne  $k$  med enere. Erstatt deretter nullene fra og med kolonne  $r'_1 + r'_2 - (k + l) + 1$  til og med kolonne  $n'$  med enere. Dette innebærer å erstatte  $(k - r'_3) + n' - (r'_1 + r'_2 - (k + l)) = n' + 2k + l - r'_1 - r'_2 - r'_3$  nullere med enere. Fra (8.6) ser vi at rad 4 fylles opp med nøyaktig  $r'_4$  enere.

**Tilfelle 2:** Anta  $r'_1 \geq k + l$ ,  $r'_2 < k + l$ .

1. Rad 2: Erstatt nullene i de  $r'_2$  første kolonnene med enere.
2. Rad 3: Erstatt nullene fra og med kolonne  $r'_2 + 1$  mot kolonne  $k + l$  med enere. Hvis  $r'_3 > (k + l) - r'_2$ : Erstatt nullene fra og med kolonne 1 mot kolonne  $k$  med enere. Hvis  $r'_3 > (k + l) - r'_2 + k$ : Erstatt nullene fra og med kolonne  $r'_1 + 1$  mot kolonne  $n'$  med enere. Med  $r_4 \geq 0$  innsatt i (8.6) følger det at  $r'_3 \leq n' + 2k + l - r'_1 - r'_2$ , som gir at det er plass til  $r'_3$  enere i tredje rad.
3. Rad 4:
  - (a) Hvis  $r'_3 \geq (k + l) - r'_2 + k$ : Erstatt nullene fra og med kolonne  $r'_1 + (r'_3 - (k + l - r'_2) - k) + 1$  til og med kolonne  $n'$  med enere. Dette innebærer å erstatte  $n' - r'_1 - (r'_3 - (k + l - r'_2) - k) = n' + 2k + l - r'_1 - r'_2 - r'_3$  nullere med enere. Fra (8.6) ser vi at rad 4 fylles opp med nøyaktig  $r'_4$  enere.
  - (b) Hvis  $(k + l) - r'_2 \leq r'_3 < (k + l) - r'_2 + k$ : Erstatt nullene fra og med kolonne  $r'_3 - (k + l - r'_2) + 1$  til og med kolonne  $k$  med enere. Erstatt videre nullene fra og med kolonne  $r'_1 + 1$  til og med kolonne  $n'$  med enere. Dette innebærer å erstatte  $k - (r'_3 - (k + l - r'_2)) + (n' - r'_1) = n' + 2k + l - r'_1 - r'_2 - r'_3$  nullere med enere. Fra (8.6) ser vi at rad 4 fylles opp med nøyaktig  $r'_4$  enere.
  - (c) Hvis  $r'_3 < k + l - r'_2$ : erstatt nullene fra og med kolonne  $r'_2 + r'_3 + 1$  til og med kolonne  $k + l$  med enere. Erstatt videre nullene i de  $k$  første kolonnene med enere og nullene fra og med kolonne  $r'_1 + 1$  til og med kolonne  $n'$  med enere. Dette innebærer å erstatte  $(k + l - r'_2 - r'_3) + k + (n' - r'_1) = n' + 2k + l - r'_1 - r'_2 - r'_3$  nullere med enere. Fra (8.6) ser vi at rad 4 fylles opp med nøyaktig  $r'_4$  enere.

**Tilfelle 3:** Anta  $r'_1 < k + l$ .

1. Rad 2: Erstatt nullene fra og med kolonne  $r'_1 + 1$  til og med kolonne  $k + l$  med enere. Med  $n' \geq l$  innsatt i (8.6), kombinert med at elementene i  $R$  er ikke-økende følger det at  $r'_1 + r'_2 \geq k + l$ . Erstatt derfor også nullene fra og med kolonne 1 mot kolonne  $k + l$  med enere.
2. Rad 3: Erstatt nullene fra og med kolonne  $r'_2 - (k + l - r'_1) + 1$  mot kolonne  $k + l$  med enere. Hvis  $r'_3 > 2(k + l) - r'_1 - r'_2$ : erstatt nullene fra og med kolonne 1 mot kolonne  $k$  med enere. Hvis  $r'_3 > 2(k + l) - r'_1 - r'_2 + k$ . Erstatt nullene fra og med kolonne  $k + l + 1$  mot kolonne  $n'$  med enere. Med  $r'_4 \geq 0$  innsatt i (8.6) følger det at  $r'_3 \leq n' + 2k + l - r'_1 - r'_2$ , som gir at det er plass til  $r'_3$  enere i tredje rad.
3. Rad 4:
  - (a) Hvis  $r'_3 \geq 2(k + l) - r'_1 - r'_2 + k$ : Erstatt nullene fra og med kolonne  $k + l + r'_3 - (2(k + l) - r'_1 - r'_2 + k) + 1$  til og med kolonne  $n'$  med enere. Dette innebærer å erstatte  $n' - (k + l) - r'_3 + (2(k + l) - r'_1 - r'_2 + k) = n' + 2k + l - r'_1 - r'_2 - r'_3$  nullere med enere. Fra (8.6) ser vi at rad 4 fylles opp med nøyaktig  $r'_4$  enere.
  - (b) Hvis  $2(k + l) - r'_1 - r'_2 \geq r'_3 < 2(k + l) - r'_1 - r'_2 + k$ : Erstatt nullene fra og med kolonne  $r'_3 - (2(k + l) - r'_1 - r'_2) + 1$  til og med kolonne  $k$  med enere. Erstatt deretter nullene fra og med kolonne  $k + l + 1$  til og med kolonne  $n'$  med enere. Dette innebærer

### 8.3. Eksistens av (0,1)-matriser med fire rader og gitte linjesummer

å erstatte  $k - r'_3 + (2(k+l) - r'_1 - r'_2) + n' - (k+l) = n' + 2k + l - r'_1 - r'_2 - r'_3$  nullere med enere Fra (8.6) ser vi at rad 4 fylles opp med nøyaktig  $r'_4$  enere.

- (c) Hvis  $r'_3 < 2(k+l) - r'_1 - r'_2$ : Erstatt elementene fra og med kolonne  $r'_1 + r'_2 + r'_3 - (k+l) + 1$  til og med kolonne  $k+l$  med enere. Erstatt deretter de  $k$  første nullene med enere og nullene fra og med kolonne  $k+l+1$  til og med kolonne  $n'$  med enere. Dette innebærer å erstatte  $2(k+l) - r'_1 - r'_2 - r'_3 + k + n' - (k+l) = n' + 2k + l - r'_1 - r'_2 - r'_3$  nullere med enere. Fra (8.6) ser vi at rad 4 fylles opp med nøyaktig  $r'_4$  enere.

For alle tilfellene får vi en  $4 \times n'$  matrise der de  $k$  første kolonnene har kolonnesum 3, de  $l$  neste har kolonnesum 2 og resten har kolonnesum 1. Videre har matrisen radsummer lik henholdsvis  $r'_1, r'_2, r'_3$  og  $r'_4$ . Ved å til slutt legge  $p$  kolonner til venstre med enere i alle rader, og  $q$  kolonner til høyre med nullere i alle rader, får vi en  $4 \times n$  (0,1)-matrise med radsumvektor  $R$  og kolonnesumvektor  $S$ . ■

Vi illustrerer algoritmen beskrevet for "Tilfelle 3 (c)" gjennom et eksempel.

**Eksempel 8.3.2.** La  $R = (8, 6, 4, 4)$  og  $S = (4, 3, 2, 2, 2, 2, 2, 2, 1, 0)$ .

Vi sjekker først at  $R$  og  $S$  oppfyller antakelsene algoritmen er basert på, det vil si (8.5a), (8.5b) og (8.5c).

$$\begin{aligned} r_1 &= 8 < n - q = 10 \quad \checkmark \\ r_1 + r_2 &= 8 + 6 = 14 < n - q + p + k + l = 19 \quad \checkmark \\ r_4 &= 4 > p = 1 \quad \checkmark. \end{aligned}$$

Fordi antakelsene er oppfylt, kan vi bruke algoritmen til å konstruere en (0,1)-matrise av størrelse  $4 \times 11$  med radsumvektor  $R$  og kolonnesumvektor  $S$ .

Fordi den første og den siste kolonnen er entydig bestemt, reduseres problemet til å finne en  $4 \times 9$ -matrise med radsumvektor  $R' = (7, 5, 3, 3)$  og kolonnesumvektor  $S' = (3, 2, 2, 2, 2, 2, 2, 2, 1)$ . Vi starter med å konstruere nullmatrisen av størrelse  $4 \times 9$ :

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

1. Rad 1: Vi erstatter nullene i de 7 første kolonnene med enere.
2. Rad 2: Vi erstatter nullen i kolonne 8 med én ener, og nullene i de 4 første kolonnene med enere.
3. Rad 3: Vi erstatter nullene fra og med kolonne 5 til og med kolonne 7 med enere.
4. Rad 4: Vi erstatter nullen i kolonne 8 med én ener, nullen i kolonne 1 med én ener og nullen i kolonne 9 med én ener.

Totalt resulterer endringene i rad 1, 2, 3 og 4, i matrisen

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

Til slutt legger vi til én kolonne til venstre med enere i alle rader, og én kolonne til høyre med nullere i alle rader. Det gir matrisen

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}.$$

△

## 8.4 Diskusjon av egne resultater mot Gale-Ryser teorem

Vi skal i dette delkapittelet vise at Teorem 8.1.1, Teorem 8.2.1 og Teorem 8.3.1 kan betraktes som forenklinger av Gale-Ryser teorem (Teorem 6.2.1) for (0,1)-matriser med to, tre og fire rader. Måten vi skal gjøre det på, er å vise at dersom  $R$  og  $S$  oppfyller (8.1), (8.3) eller (8.5) for  $m$  lik henholdsvis 2, 3 eller 4, så er  $S$  majorisert av  $R^*$ .

### Gale-Ryser versus Teorem 8.1.1

**Proposisjon 8.4.1.** *La  $R = (r_1, r_2)$  og  $S = (s_1, s_2, \dots, s_n)$  være to ikke-økende, ikke-negative heltallsvektorer med*

$$\sum_{j=1}^n s_j = r_1 + r_2.$$

*Hvis*

$$r_1 \leq n - q, \tag{8.1}$$

*så er  $S \preceq R^*$ .*

*Bevis.* Anta at  $R$  og  $S$  oppfyller (8.1). Med (8.1) innsatt i

$$r_1 + r_2 = \sum_{j=i}^n s_j = n - q + p,$$

følger det at  $r_2 \geq p$ .  $S$  og  $R^*$  kan derfor presenteres på formen

$$\begin{aligned} S &= (\underbrace{2, \dots, 2}_p, \underbrace{1, 1, 1, 1, \dots, 1, 1}_{n-q}, 0, \dots, 0), \\ R^* &= (\underbrace{2, 2, 2, \dots, 2}_{r_2}, \underbrace{1, \dots, 1}_{r_1}). \end{aligned} \tag{8.7}$$

Fra (8.7) ser vi at de  $r_2$  første elementene av  $S$  er majorisert av de  $r_2$  første elementene av  $R^*$ . Fordi  $r_1 \leq n - q$ , har vi videre  $s_j = r_j^* = 1$  for  $r_2 < j \leq r_1$ , og kan konkludere med at de  $r_1$  første elementene av  $S$  er majorisert av  $R^*$ . Til slutt har vi fra

$$\sum_{i=1}^{r_1} r_i^* = \sum_{i=1}^2 r_i = \sum_{j=1}^n s_j,$$

at de  $n - q - r_1$  siste enerne av  $S$  ikke gjør summen av  $S$  større enn summen av  $R^*$ , og resultatet følger. ■



## Gale-Ryser versus Teorem 8.2.1

Proposisjon 8.4.2. La  $R = (r_1, r_2, r_3)$  og  $S = (s_1, s_2, \dots, s_n)$  være to ikke-økende, ikke-negative heltallsvektorer med

$$\sum_{j=1}^n s_j = r_1 + r_2 + r_3.$$

Hvis

$$r_1 \leq n - q, \quad (8.3a)$$

$$r_3 \geq p, \quad (8.3b)$$

så er  $S \preceq R^*$ .

Bevis. Anta at  $R$  og  $S$  oppfyller (8.3).  $S$  og  $R^*$  kan da presenteres på formen

$$\begin{aligned} S &= (\underbrace{3, \dots, 3}_p, \underbrace{2, \dots, 2}_l, \overbrace{1, 1, 1, \dots, 1, 1}^{n-q}, 0, \dots, 0), \\ R^* &= (\underbrace{3, 3, \dots, 3}_{r_3}, \overbrace{2, 2, \dots, 2}^{r_1}, \underbrace{1, \dots, 1}_{r_2}). \end{aligned} \quad (8.8)$$

I (8.8) er (8.3a) og (8.3b) riktig fremstilt, mens  $r_2$  kan være mindre, lik eller større enn  $p+l$ .

Fordi  $r_3 \geq p$ , følger det at de  $r_3$  første elementene av  $S$  er majorisert av de  $r_3$  første elementene av  $R^*$ . Videre kan vi ha to tilfeller.

1. Hvis  $r_2 \geq p+l$ , ser vi fra (8.8) at de  $r_2$  første elementene av  $S$  er majorisert av de  $r_2$  første elementene i  $R^*$ . Fordi  $r_1 \leq n - q$ , har vi videre  $s_j = r_j^* = 1$  for  $r_2 < j \leq r_1$  og kan konkludere med at de  $r_1$  første elementene av  $S$  er majorisert av  $R^*$ .
2. Hvis  $r_2 < p+l$ , så er  $s_j = r_j^* = 2$  for  $r_3 < j \leq r_2$ , mens  $s_j = 2$  og  $r_j^* = 1$  for  $r_2 < j \leq p+l$ . Vi vet derfor at dersom de  $p+l$  første elementene av  $S$  er majorisert av de  $p+l$  første elementene av  $R^*$ , så er også de  $j$  første elementene av  $S$  er majorisert av de  $j$  første elementene av  $R^*$ , for  $j < p+l$ .

$$\sum_{j=1}^{p+l} r_j^* - \sum_{j=1}^{p+l} s_j = 3r_3 + 2(r_2 - r_3) + (p+l - r_2) - (3p+2l) = r_3 + r_2 - 2p - l \geq 0.$$

Siste ulikhet følger fra  $r_1 \leq n - q$  innsatt i

$$r_1 + r_2 + r_3 = \sum_{j=1}^n s_n = n - q + 2p + l.$$

Fordi  $r_1 \leq n - q$ , har vi videre  $s_k = r_k^* = 1$  for  $p+l < j \leq r_1$  og kan konkludere med at de  $r_1$  første elementene av  $S$  er majorisert av  $R^*$ .

For begge tilfellene, har vi til slutt fra

$$\sum_{i=1}^{r_1} r_i^* = \sum_{i=1}^3 r_i = \sum_{j=1}^n s_j,$$

at de  $n - q - r_1$  siste enerne av  $S$  ikke gjør summen av  $S$  større enn summen av  $R^*$ , og resultatet følger. ■

## Gale-Ryser versus Teorem 8.3.1

Proposisjon 8.4.3. La  $R = (r_1, r_2, r_3, r_4)$  og  $S = (s_1, s_2, \dots, s_n)$  være to ikke-økende, ikke-negative heltallsvektorer med

$$\sum_{j=1}^n s_j = r_1 + r_2 + r_3 + r_4.$$

Hvis

$$r_1 \leq n - q, \quad (8.5a)$$

$$r_4 \geq p, \quad (8.5b)$$

$$r_1 + r_2 \leq n - q + p + k + l, \quad (8.5c)$$

så er  $S \preceq R^*$ .

Bevis. Anta at  $R$  og  $S$  oppfyller (8.5).  $S$  og  $R^*$  kan da presenteres på formen

$$\begin{aligned} S &= (\overbrace{4, \dots, 4, 3, \dots, 3, 2, \dots, 2}^{n-q}, 1, 1, 1, 1, 1, \dots, 1, 0, \dots, 0), \\ R^* &= (\overbrace{4, 4, \dots, 4, 3, 3, \dots, 3}^{r_1}, \overbrace{2, 2, \dots, 2}^{r_4}, \overbrace{1, \dots, 1}^{r_3}). \end{aligned} \quad (8.9)$$

I (8.9) er (8.5a) og (8.5c) riktig fremstilt, mens forholdet mellom  $r_3$  og  $p + k$  og mellom  $r_2$  og  $p + k + l$  kan variere.

Fordi  $r_4 \geq p$ , følger det at de  $r_4$  første elementene av  $S$  er majorisert av de  $r_4$  første elementene av  $R^*$ . Videre kan vi ha ulike tilfeller.

1. Hvis  $r_3 \geq p + k$  og  $r_2 \geq p + k + l$ , ser vi fra (8.9) at de  $r_2$  første elementene av  $S$  er majorisert av de  $r_2$  første elementene av  $R^*$ . Fordi  $r_1 \leq n - q$ , har vi videre  $s_j = r_j^* = 1$  for  $r_2 < j \leq r_1$ , og kan konkludere med at de  $r_1$  første elementene av  $S$  er majorisert av  $R^*$ .
2. Hvis  $r_3 \geq p + k$  og  $r_2 < p + k + l$ , ser vi fra (8.9) at de  $r_3$  første elementene av  $S$  er majorisert av de  $r_3$  første elementene av  $R^*$ . Videre er  $s_j = r_j^* = 2$  for  $r_3 < j \leq r_2$ , mens  $s_j = 2$  og  $r_j^* = 1$  for  $r_2 < j \leq p + k + l$ . Vi vet derfor at dersom de  $p + k + l$  første elementene av  $S$  er majorisert av de  $p + k + l$  første elementene av  $R^*$ , så er også de  $j$  første elementene av  $S$  majorisert av de  $j$  første elementene av  $R^*$ , for  $j < p + k + l$ .

$$\begin{aligned} \sum_{j=1}^{p+k+l} r_j^* - \sum_{j=1}^{p+k+l} s_j &= 4r_4 + 3(r_3 - r_4) + 2(r_2 - r_3) + (p + k + l - r_2) - (4p + 3k + 2l) \\ &= r_4 + r_3 + r_2 - 3p - 2k - l \geq 0. \end{aligned}$$

Siste ulikhet følger fra  $r_1 \leq n - q$  innsatt i

$$r_1 + r_2 + r_3 + r_4 = \sum_{i=1}^n s_n = n - q + 3p + 2k + l.$$

Fordi  $r_1 \leq n - q$ , har vi videre  $s_j = r_j^* = 1$  for  $p + k + l < j \leq r_1$ , og kan konkludere med at de  $r_1$  første elementene av  $S$  er majorisert av  $R^*$ .

## 8.5. MATLAB-program for å lage (0,1)-matriser med to, tre og fire rader og gitte linjesummer

3. Hvis  $r_3 < p + k$ , så er  $s_j = r_j^* = 3$  for  $r_4 < j \leq r_3$ , mens  $s_j = 3$  og  $r_j^* = 2$  for  $r_3 < j \leq p + k$ . Vi vet derfor at dersom de  $p + k$  første elementene av  $S$  er majorisert av de  $p + k$  første elementene av  $R^*$ , så er også de  $j$  første elementene av  $S$  er majorisert av de  $j$  første elementene av  $R^*$ , for  $j < p + k$ .

$$\begin{aligned} \sum_{j=1}^{p+k} r_j^* - \sum_{j=1}^{p+k} s_j &= 4r_4 + 3(r_3 - r_4) + 2(p + k - r_3) - 4p - 3k \\ &= r_4 + r_3 - 2p - k \geq 0 \end{aligned}$$

Siste ulikhet følger fra  $r_1 + r_2 \leq n - q + p + k + l$  innsatt i

$$r_1 + r_2 + r_3 + r_4 = \sum_{i=1}^n s_n = n - q + 3p + 2k + l.$$

- (a) Hvis  $r_2 \geq p + k + l$ , så er  $s_j = r_j^* = 2$  for  $p + k < j \leq p + k + l$ , mens  $s_j = 1$  og  $r_j^* = 2$  for  $p + k + l < j \leq r_2$ . Derfor er de  $r_2$  første elementene av  $S$  er majorisert av de  $r_2$  første elementene av  $R^*$ . Fordi  $r_1 \leq n - q$ , har vi  $s_j = r_j^* = 1$  for  $r_2 < j \leq r_1$ , og kan konkludere med at de  $r_1$  første elementene av  $S$  er majorisert av  $R^*$ .
- (b) Hvis  $r_2 < p + k + l$  gir samme resonnement som beskrevet for punkt 2. at de  $r_1$  første elementene av  $S$  er majorisert av  $R^*$ .

For alle de tre tilfellene, har vi til slutt fra

$$\sum_{i=1}^{r_1} r_i^* = \sum_{i=1}^4 r_i = \sum_{j=1}^n s_j,$$

at de  $n - q - r_1$  siste enerne av  $S$  ikke gjør summen av  $S$  større enn summen av  $R^*$ , og resultatet følger. ■

## 8.5 MATLAB-program for å lage (0,1)-matriser med to, tre og fire rader og gitte linjesummer

Basert på konstruksjonen av en (0,1)-matrise med gitte linjesummer  $R$  og  $S$ , formulert i bevisene av Teorem 8.1.1, Teorem 8.2.1 og Teorem 8.3.1, har jeg brukt MATLAB [MAT22] til å skrive et program som returnerer en slik matrise gitt at det eksisterer. Nedenfor presenteres koden i korte trekk, mens koden i sin helhet med beskrivelser er gitt i Tillegg C.

Koden består av en hovedfunksjon (gitt i Tillegg C.4) som tar to vektorer  $R$  og  $S$  som argumenter. Funksjonen sjekker først om summen av elementene i  $R$  er lik summen av elementene i  $S$ . Dersom de er like bestemmes lengden av  $R$ .

- Dersom  $R$  består av to elementer, kaller hovedfunksjonen på en egendefinert funksjon (gitt i Tillegg C.1). Denne funksjonen bestemmer først antall elementer i  $S$  som er lik henholdsvis 0 og 2. Deretter sjekker funksjonen om  $R$  og  $S$  oppfyller (8.1). Dersom ulikheten oppfylles, returnerer funksjonen en (0,1)-matrise med to rader, radsumvektor  $R$  og kolonnesumvektor  $S$ .

## Kapittel 8. Eksistens av (0,1)-matriser med to, tre og fire rader og gitte linjesummer

- Dersom  $R$  består av tre elementer, kaller hovedfunksjonen på en egendefinert funksjon (gitt i Tillegg C.2). Denne funksjonen bestemmer først antall elementer i  $S$  som er lik henholdsvis 0, 2 og 3. Deretter sjekker funksjonen om  $R$  og  $S$  oppfyller (8.3a) og (8.3b). Dersom begge ulikhetene oppfylles, returnerer funksjonen en (0,1)-matrise med tre rader, radsumvektor  $R$  og kolonnesumvektor  $S$ .
- Dersom  $R$  består av fire elementer, kaller hovedfunksjonen på en egendefinert funksjon (gitt i Tillegg C.3). Denne funksjonen bestemmer først antall elementer i  $S$  som er lik henholdsvis 0, 2, 3 og 4. Deretter sjekker funksjonen om  $R$  og  $S$  oppfyller (8.5a), (8.5b) og (8.5c). Dersom alle ulikhetene oppfylles, returnerer funksjonen en (0,1)-matrise med fire rader, radsumvektor  $R$  og kolonnesumvektor  $S$ .
- Dersom lengden av  $R$  er forskjellig fra to, tre eller fire, returnerer funksjonen en kommentar om at det.

I koden anvendes de innebygde MATLAB-funksjonene *length*, *zeros* og *sum* [MAT22].

**Eksempel 8.5.1.** La oss anvende koden til å bestemme en (0,1)-matrise med radsumvektor  $R = (8, 6, 4, 4)$  og kolonnesumvektor  $S = (4, 3, 2, 2, 2, 2, 2, 2, 2, 1, 0)$ .

```
>> M=matrisefunksjon(R,S)
```

M =

```
1     1     1     1     1     1     1     1     0     0     0
1     1     1     1     1     0     0     0     1     0     0
1     0     0     0     0     1     1     1     0     0     0
1     1     0     0     0     0     0     0     1     1     0
```

Vi ser at matrisen som returneres er den samme som ble bestemt i Eksempel 8.3.2.  $\triangle$

## Kapittel 9

# Konklusjon

Vi har i denne oppgaven fått et lite innblikk i ulike nettverk strøm problemer og anvendelser av disse. Vi har blant beskrevet hvordan vi kan løse et tillatt strøm problem for å avgjøre om det eksisterer, og i så fall finne en matrise med gitte linjesummer. Videre har vi sett hvordan vi kan tilordne enkelte kanter i et nettverk øvre kapasiteter, for å sikre at bestemte elementer i den korresponderende matrisen oppfyller gitte betingelser. I oppgavens utforskende del (kapittel 7 og 8) utnytter vi denne sammenhengen til å avgjøre om det eksisterer, og i så fall finne matriser med bestemte strukturer og gitte linjesummer. I disse kapitlene har vi også undersøkt hvordan vi kan avgjøre når denne typen matriser eksisterer, uten å løse et tillatt strømproblem. I kapittel 7 presenteres og bevises resultater som karakteriserer når det eksisterer en ikke-negativ nedre triangulær, øvre triangulær og en tridiagonal matrise med gitte linjesummer. Vi viser i tillegg at disse resultatene kan betraktes som spesielle tilfeller av “Teorem 4” i [Dah13]. I kapittel 8 presenteres og bevises resultater som karakteriserer når det eksisterer en  $(0, 1)$ -matrise med gitte linjesummer, når vi begrenser antall rader til to, tre eller fire. Avslutningsvis viser vi at disse resultatene kan betraktes som forenklinger av Gale-Ryser teorem for  $(0, 1)$ -matriser med to, tre og fire rader.

Da samtlige av egne resultater, allerede er bevist for mer generelle tilfeller, er det begrenset hvor interessant det er for andre å bygge videre på diskusjonen i denne oppgaven. Det må imidlertid nevnes at det å utlede og oppdage sammenhenger, uten å være begrenset av at sammenhengene allerede er bevist, har vært en lærerrik prosess. Med dette perspektivet kan det derfor likevel være interessant for andre å fortsette diskusjonen, for eksempel for å bevise resultatene for nedre triangulære, øvre triangulære og tridiagonale matriser uten å bruke [Dah13] sitt teorem. Videre kan det være interessant å utforske hvordan andre mønstre enn trappemønstre påvirker hvorvidt det eksisterer en ikke-negativ  $m \times n$  matrise med gitte linjesummer.

Uavhengig av om denne oppgaven er utgangspunkt for videre diskusjon, håper jeg den kan inspirere leseren til å ha en utforskende holdning til matematikken med mål om å selv oppdage sammenhenger. Jeg sitter igjen med en erfaring av at prosessen ble målet.

## Kapittel 9. Konklusjon

# Referanser

- [AMO93] Ahuja, R., Magnanti, T. og Orlin, J. B. *Network Flows: Theory, Algorithms, and Applications*. Englewood Cliffs, New Jersey: Prentice-Hall, 1993.
- [BM08] Bondy, J. A. og Murty, U. S. R. *Graph theory*. Springer, 2008.
- [Bru06] Brualdi, R. A. *Combinatorial Matrix Classes*. Cambridge University Press, 2006.
- [Dah08] Dahl, G. «Transportation matrices with staircase patterns and majorization». I: *Linear Algebra and its Applications* 429 (okt. 2008), s. 1840–1850.
- [Dah13] Dahl, G. *Network flows and combinatorial matrix theory*. Lecture notes. 2013.
- [Kra96] Krause, M. «A Simple Proof of the Gale-Ryser Theorem». I: *The American Mathematical Monthly* 103.4 (1996), s. 335–337.
- [MAT22] MATLAB. *version 9.13.0 (R2022b)*. Natick, Massachusetts: The MathWorks Inc., 2022.

## Referanser



## Tillegg A

# Bevis tilhørende kapittel 7

### A.1 Nordvest-Hjørne regelen produserer nedre triangulære transportmatriser

Anta at to vektorer  $R$  og  $S$  oppfyller (7.1) for  $n = 3$ . For å vise at Nordvest-Hjørne regelen produserer en nedre triangulær  $3 \times 3$  matrise  $A = [a_{ij}]$  med radsumvektor  $R$  og kolonnesumvektor  $S$ , må vi vise at  $a_{12} = a_{13} = a_{23} = 0$ .

Fra  $r_1 \leq s_1$  følger det at  $a_{11} = r_1$  og  $a_{12} = a_{13} = 0$ . Videre kan vi ha to tilfeller.

1. Hvis  $r_2 \leq s_1 - r_1$ , gir regelen  $a_{21} = r_2$  og  $a_{22} = a_{23} = 0$ .
2. Hvis  $r_2 > s_1 - r_1$ , gir regelen  $a_{21} = s_1 - r_1$  og  $a_{31} = 0$ . Fordi  $r_2 + r_1 - s_1 \leq s_2$  følger det videre at  $a_{22} = r_2 + r_1 - s_1$  og  $a_{23} = 0$ .

Anta at to vektorer  $R$  og  $S$  oppfyller (7.1) for  $n = 4$ . For å vise at Nordvest-Hjørne regelen produserer en nedre triangulær  $4 \times 4$  matrise  $A = [a_{ij}]$  med radsumvektor  $R$  og kolonnesumvektor  $S$ , må vi vise at  $a_{12} = a_{13} = a_{14} = a_{23} = a_{24} = a_{34} = 0$ .

Fra  $r_1 \leq s_1$  følger det at  $a_{11} = r_1$  og  $a_{12} = a_{13} = a_{14} = 0$ . Videre kan vi ha ulike tilfeller

1. Hvis  $r_1 + r_2 + r_3 \leq s_1$ , gir regelen  $a_{21} = r_2$ ,  $a_{31} = r_3$  og at resten av elementene i rad to og tre er null.
2. Hvis  $r_1 + r_2 \leq s_1$ , mens  $r_3 > s_1 - r_1 - r_2$ , gir første ulikhet  $a_{21} = r_2$  og at resten av elementene i rad to er null. Videre gir andre ulikhet at  $a_{31} = s_1 - r_1 - r_2$  og  $a_{41} = 0$ .
  - (a) Hvis  $r_3 + r_2 + r_1 - s_1 \leq s_2$ , gir regelen  $a_{32} = r_3 + r_2 + r_1 - s_1$  og  $a_{33} = a_{34} = 0$ .
  - (b) Hvis  $r_3 + r_2 + r_1 - s_1 > s_2$ , gir regelen  $a_{32} = s_2$  og  $a_{42} = 0$ . Fordi  $r_3 + r_2 + r_1 - s_1 - s_2 \leq s_3$  følger det videre at  $a_{33} = r_3 + r_2 + r_1 - s_1 - s_2$  og  $a_{34} = 0$ .
3. Hvis  $r_2 > s_1 - r_1$ , gir regelen  $a_{21} = s_1 - r_1$  og  $a_{31} = a_{41} = 0$ . Fordi  $r_2 + r_1 - s_1 \leq s_2$  følger det videre at  $a_{22} = r_2 + r_1 - s_1$  og  $a_{23} = a_{24} = 0$ .
  - (a) Hvis  $r_3 \leq s_2 + s_1 - r_1 - r_2$ , gir regelen  $a_{32} = r_3$  og  $a_{33} = a_{34} = 0$ .

- (b) Hvis  $r_3 > s_2 + s_1 - r_1 - r_2$ , gir regelen  $a_{32} = s_2 + s_1 - r_1 - r_2$  og  $a_{42} = 0$ . Fordi  $r_3 + r_2 + r_1 - s_1 - s_2 \leq s_3$  følger det videre at  $a_{33} = r_3 + r_2 + r_1 - s_1 - s_2$  og  $a_{34} = 0$ .

Anta at to vektorer  $R$  og  $S$  oppfyller (7.1) for  $n = 5$ . For å vise at Nordvest-Hjørne regelen produserer en nedre triangulær  $5 \times 5$  matrise  $A = [a_{ij}]$  med radsumvektor  $R$  og kolonnesumvektor  $S$ , må vi vise at  $a_{12} = a_{13} = a_{14} = a_{15} = a_{23} = a_{24} = a_{25} = a_{34} = a_{35} = a_{45} = 0$ .

Fra  $r_1 \leq s_1$  følger det at  $a_{11} = r_1$  og  $a_{12} = a_{13} = a_{14} = a_{15} = 0$ . Videre kan vi ha ulike tilfeller.

1. Hvis  $r_1 + r_2 + r_3 + r_4 \leq s_1$ , gir regelen  $a_{21} = r_2$ ,  $a_{31} = r_3$ ,  $a_{41} = r_4$ , og at resten av elementene i rad to, tre og fire er lik null.
2. Hvis  $r_1 + r_2 + r_3 \leq s_1$ , mens  $r_4 > s_1 - r_1 - r_2 - r_3$ , gir første ulikhet at  $a_{21} = r_2$ ,  $a_{31} = r_3$ , og at resten av elementene i rad to og tre er lik null. Videre gir andre ulikhet at  $a_{41} = s_1 - r_1 - r_2 - r_3$  og  $a_{51} = 0$ .

- (a) Hvis  $r_4 + r_3 + r_2 + r_1 - s_1 \leq s_2$ , gir regelen  $a_{42} = r_4 + r_3 + r_2 + r_1 - s_1$  og  $a_{43} = a_{44} = a_{45} = 0$ .

- (b) Hvis  $r_1 + r_2 + r_3 + r_4 - s_1 > s_2$ , gir regelen  $a_{42} = s_2$  og  $a_{52} = 0$ .

- i. Hvis  $r_4 + r_3 + r_2 + r_1 - s_1 - s_2 \leq s_3$ , gir regelen  $a_{43} = r_4 + r_3 + r_2 + r_1 - s_1 - s_2$  og  $a_{44} = a_{45} = 0$ .

- ii. Hvis  $r_4 + r_3 + r_2 + r_1 - s_1 - s_2 > s_3$ , gir regelen  $a_{43} = s_3$  og  $a_{53} = 0$ . Fordi  $r_4 + r_3 + r_2 + r_1 - s_1 - s_2 - s_3 \leq s_4$  følger det videre at  $a_{44} = r_4 + r_3 + r_2 + r_1 - s_1 - s_2 - s_3$  og  $a_{45} = 0$ .

3. Hvis  $r_1 + r_2 \leq s_1$ , mens  $r_3 > s_1 - r_1 - r_2$  gir første ulikhet at  $a_{21} = r_2$  og at resten av elementene i rad to er null. Videre gir andre ulikhet at  $a_{31} = s_1 - r_1 - r_2$  og  $a_{41} = a_{51} = 0$ .

- (a) Hvis  $r_1 + r_2 + r_3 - s_1 \leq s_2$ , gir regelen  $a_{32} = r_3 + r_2 + r_1 - s_1$  og  $a_{33} = a_{34} = a_{35} = 0$ .

- i. Hvis  $r_4 \leq s_1 + s_2 - r_1 - r_2 - r_3$ , gir regelen  $a_{42} = r_4$  og  $a_{43} = a_{44} = a_{45} = 0$ .

- ii. Hvis  $r_4 > s_2 + s_1 - r_1 - r_2 - r_3$ , gir regelen  $a_{42} = s_2 + s_1 - r_1 - r_2 - r_3$  og  $a_{52} = 0$ .

- A. Hvis  $r_4 + r_3 + r_2 + r_1 - s_1 - s_2 \leq s_3$ , gir regelen  $a_{43} = r_4 + r_3 + r_2 + r_1 - s_1 - s_2$  og  $a_{44} = a_{45} = 0$ .

- B. Hvis  $r_4 + r_3 + r_2 + r_1 - s_1 - s_2 > s_3$ , gir regelen  $a_{43} = s_3$  og  $a_{53} = 0$ . Fordi  $r_4 + r_3 + r_2 + r_1 - s_1 - s_2 - s_3 \leq s_4$  følger det videre at  $a_{44} = r_4 + r_3 + r_2 + r_1 - s_1 - s_2 - s_3$  og  $a_{45} = 0$ .

- (b) Hvis  $r_3 + r_2 + r_1 - s_1 > s_2$ , gir regelen  $a_{32} = s_2$  og  $a_{42} = a_{52} = 0$ . Fordi  $r_3 + r_2 + r_1 - s_1 - s_2 \leq s_3$  følger det videre at  $a_{33} = r_3 + r_2 + r_1 - s_1 - s_2$  og  $a_{34} = a_{35} = 0$ .

- i. Hvis  $r_4 \leq s_3 + s_2 + s_1 - r_1 - r_2 - r_3$ , gir regelen  $a_{43} = r_4$  og  $a_{44} = a_{45} = 0$ .

### A.1. Nordvest-Hjørne regelen produserer nedre triangulære transportmatriser

- ii. Hvis  $r_4 > s_3 + s_2 + s_1 - r_1 - r_2 - r_3$ , gir regelen  $a_{43} = s_3 + s_2 + s_1 - r_1 - r_2 - r_3$  og  $a_{53} = 0$ . Fordi  $r_4 + r_3 + r_2 + r_1 - s_1 - s_2 - s_3 \leq s_4$  følger det videre at  $a_{44} = r_4 + r_3 + r_2 + r_1 - s_1 - s_2 - s_3$  og  $a_{45} = 0$ .
4. Hvis  $r_2 > s_1 - r_1$ , gir regelen  $a_{21} = s_1 - r_1$  og  $a_{31} = a_{41} = a_{51} = 0$ . Fordi  $r_2 + r_1 - s_1 \leq s_2$  følger det videre at  $a_{22} = r_2 + r_1 - s_1$  og  $a_{23} = a_{24} = a_{25} = 0$ .
- (a) Hvis  $r_3 \leq s_2 + s_1 - r_1 - r_2$ , gir regelen  $a_{32} = r_3$  og  $a_{34} = a_{35} = 0$ .
- i. Hvis  $r_4 \leq s_2 + s_1 - r_1 - r_2 - r_3$ , gir regelen  $a_{42} = r_4$  og  $a_{43} = a_{44} = a_{45} = 0$ .
- ii. Hvis  $r_4 > s_2 + s_1 - r_1 - r_2 - r_3$ , gir regelen  $a_{42} = s_2 + s_1 - r_1 - r_2 - r_3$  og  $a_{52} = 0$ .
- A. Hvis  $r_4 + r_3 + r_2 + r_1 - s_1 - s_2 \leq s_3$ , gir regelen  $a_{43} = r_4 + r_3 + r_2 + r_1 - s_1 - s_2$  og  $a_{44} = a_{45} = 0$ .
- B. Hvis  $r_4 + r_3 + r_2 + r_1 - s_1 - s_2 > s_3$ , gir regelen  $a_{43} = s_3$  og  $a_{53} = 0$ . Fordi  $r_4 + r_3 + r_2 + r_1 - s_1 - s_2 - s_3 \leq s_4$ , følger det videre at  $a_{44} = r_4 + r_3 + r_2 + r_1 - s_1 - s_2 - s_3$  og  $a_{45} = 0$ .
- (b) Hvis  $r_3 > s_2 + s_1 - r_1 - r_2$ , gir regelen  $a_{32} = s_2 + s_1 - r_1 - r_2$  og  $a_{42} = a_{52} = 0$ . Fordi  $r_3 + r_2 + r_1 - s_1 - s_2 \leq s_3$  følger det videre at  $a_{33} = r_3 + r_2 + r_1 - s_1 - s_2$  og  $a_{34} = a_{35} = 0$ .
- i. Hvis  $r_4 \leq s_3 + s_2 + s_1 - r_1 - r_2 - r_3$ , gir regelen  $a_{43} = r_4$  og  $a_{44} = a_{45} = 0$ .
- ii. Hvis  $r_4 > s_3 + s_2 + s_1 - r_1 - r_2 - r_3$ , gir regelen  $a_{43} = s_3 + s_2 + s_1 - r_1 - r_2 - r_3$  og  $a_{53} = 0$ . Fordi  $r_4 + r_3 + r_2 + r_1 - s_1 - s_2 - s_3 \leq s_4$ , følger det videre at  $a_{44} = r_4 + r_3 + r_2 + r_1 - s_1 - s_2 - s_3$  og  $a_{45} = 0$ .

### Induksjonsbevis tilhørende delkapittel 7.2

Viser ved induksjon at dersom to vektorer  $R = (r_1, \dots, r_n)$  og  $S = (s_1, \dots, s_n)$  oppfyller (7.1) og (7.2), så produserer Nordvest-hjørne regelen en matrise  $A \in \mathcal{N}_W(R, S)$ , der  $W$  er trappematriksen med  $v_1 = 1$ ,  $v_i = i - 1$  for  $i = 2, \dots, n$  og  $h_i = i$  for  $i = 1, \dots, n$ .

*Bevis.* Anta at  $R$  og  $S$  oppfyller (7.1) og (7.2). Ved å bruke Nordvest-hjørne regelen til å fullføre elementene til og med rad 3 får vi delmatrisen:

$$\begin{bmatrix} r_1 & 0 & 0 & 0 & \dots & 0 \\ s_1 - r_1 & r_2 + r_1 - s_1 & 0 & 0 & \dots & 0 \\ 0 & s_2 + s_1 - r_1 - r_2 & r_3 + r_2 + r_1 - s_1 - s_2 & 0 & \dots & 0 \\ 0 & 0 & & & & \\ \vdots & \vdots & & & & \\ 0 & 0 & & & & \end{bmatrix}$$

Anta videre at det å bruke Nordvest-hjørne regelen til å fullføre elementene til og med kolonne  $p - 1$  og rad  $p$ , gir samme mønster. Det vil si at for  $k = 2, \dots, p$ , så er:

- $a_{k,k-1} = s_{k-1} + \dots + s_1 - r_1 - \dots - r_{k-1}$ , mens  $a_{j,k-1} = 0$  for  $j > k$ .
- $a_{kk} = r_k + \dots + r_1 - s_1 - \dots - s_{k-1}$ , mens  $a_{k,j} = 0$  for  $j > k$ .

Gitt denne antakelsen skal vi vise at Nordvest-hjørne regelen også gir kolonne  $p$  og rad  $p + 1$  på samme form. Neste element som skal bestemmes er  $a_{p+1,p}$ . Fra antakelsen følger det at alle elementer i rad  $p + 1$  som står til venstre for  $a_{p+1,p}$  er lik null, og at alle elementer i kolonne  $p$  som står over  $a_{pp}$  er lik null. Vi sammenligner derfor  $r_{p+1}$  med følgende oppdaterte kolonnesum for kolonne  $p$ .

$$s'_p = s_p - a_{pp} = s_p - (r_p + \dots + r_1 - s_1 - \dots - s_{p-1}) = s_p + \dots + s_1 - r_1 - \dots - r_p.$$

Fra (7.2) har vi at  $s'_p \leq r_{p+1}$ . Nordvest-hjørne regelen gir dermed  $a_{p+1,p} = s'_p$ , mens  $a_{jp} = 0$  for  $j > p + 1$ . Neste element som skal bestemmes er derfor  $a_{p+1,p+1}$ . Fra antakelsen følger det at alle elementer i kolonne  $p + 1$  som står over  $a_{p+1,p+1}$  er lik null. Vi sammenligner derfor  $s_{p+1}$ , med følgende oppdaterte radsummen for rad  $p + 1$ .

$$r'_{p+1} = r_{p+1} - a_{p+1,p} = r_{p+1} - (s_p + \dots + s_1 - r_1 - \dots - r_p) = r_{p+1} + \dots + r_1 - s_1 - \dots - s_p.$$

Fra (7.1) har vi at  $r'_{p+1} \leq s_{p+1}$ . Nordvest-hjørne regelen gir dermed at  $a_{p+1,p+1} = r'_{p+1}$ , mens  $a_{p+1,j} = 0$  for  $j > p + 1$ . ■

## A.2 Nordvest-Hjørne regelen produserer øvre triangulære transportmatriser

Anta at to vektorer  $R$  og  $S$  oppfyller (7.3) for  $n = 3$ . For å vise at Nordvest-Hjørne regelen produserer en øvre triangulær  $3 \times 3$  matrise  $A = [a_{ij}]$  med radsumvektor  $R$  og kolonnesumvektor  $S$ , må vi vise at  $a_{21} = a_{31} = a_{32} = 0$ .

Fra  $s_1 \leq r_1$  følger det at  $a_{11} = s_1$  og  $a_{21} = a_{31} = 0$ . Videre kan vi ha to tilfeller.

1. Hvis  $r_1 - s_1 \geq s_2$ , gir regelen  $a_{12} = s_2$  og  $a_{22} = a_{32} = 0$ .
2. Hvis  $r_1 - s_1 < s_2$ , gir regelen  $a_{12} = r_1 - s_1$  og  $a_{13} = 0$ . Fordi  $s_2 + s_1 - r_1 \leq r_2$  følger det videre at  $a_{22} = s_2 + s_1 - r_1$  og  $a_{32} = 0$ .

Anta at to vektorer  $R$  og  $S$  oppfyller (7.3) for  $n = 4$ . For å vise at Nordvest-Hjørne regelen produserer en øvre triangulær  $4 \times 4$  matrise  $A = [a_{ij}]$  med radsumvektor  $R$  og kolonnesumvektor  $S$ , må vi vise at  $a_{21} = a_{31} = a_{41} = a_{32} = a_{42} = a_{43} = 0$ .

Fra  $s_1 \leq r_1$  følger det at  $a_{11} = s_1$  og  $a_{21} = a_{31} = a_{41} = 0$ . Videre kan vi ha ulike tilfeller.

1. Hvis  $r_1 \geq s_1 + s_2 + s_3$ , gir regelen  $a_{12} = s_2$ ,  $a_{13} = s_3$  og at resten av elementene i kolonne to og tre er null.
2. Hvis  $r_1 \geq s_1 + s_2$ , mens  $r_1 - s_1 - s_2 < s_3$ , gir første ulikhet  $a_{12} = s_2$  og at resten av elementene i kolonne to er null. Videre gir andre ulikhet  $a_{13} = r_1 - s_1 - s_2$  og  $a_{14} = 0$ .
  - (a) Hvis  $r_2 \geq s_3 + s_2 + s_1 - r_1$ , gir regelen  $a_{23} = s_3 + s_2 + s_1 - r_1$  og  $a_{33} = a_{43} = 0$ .
  - (b) Hvis  $r_2 < s_3 + s_2 + s_1 - r_1$ , gir regelen  $a_{23} = r_2$  og  $a_{24} = 0$ . Fordi  $s_3 + s_2 + s_1 - r_1 - r_2 \leq r_3$  følger det videre at  $a_{33} = s_3 + s_2 + s_1 - r_1 - r_2$  og  $a_{43} = 0$ .

## A.2. Nordvest-Hjørne regelen produserer øvre triangulære transportmatriser

3. Hvis  $r_1 - s_1 < s_2$ , gir regelen  $a_{12} = r_1 - s_1$  og  $a_{13} = a_{14} = 0$ . Fordi  $s_2 + s_1 - r_1 \leq r_2$  følger det videre at  $a_{22} = s_2 + s_1 - r_1$  og  $a_{32} = a_{42} = 0$ .

(a) Hvis  $r_2 + r_1 - s_1 - s_2 \geq s_3$ , gir regelen  $a_{23} = s_3$  og  $a_{33} = a_{43} = 0$ .

(b) Hvis  $r_2 + r_1 - s_1 - s_2 < s_3$ , gir regelen  $a_{23} = r_2 + r_1 - s_1 - s_2$  og  $a_{24} = 0$ . Fordi  $s_3 + s_2 + s_1 - r_1 - r_2 \leq r_3$  følger det videre at  $a_{33} = s_3 + s_2 + s_1 - r_1 - r_2$  og  $a_{43} = 0$ .

Anta at to vektorer  $R$  og  $S$  oppfyller (7.3) for  $n = 5$ . For å vise at Nordvest-Hjørne regelen produserer en øvre triangulær  $5 \times 5$  matrise  $A = [a_{ij}]$  med radsumvektor  $R$  og kolonnesumvektor  $S$ , må vi vise at  $a_{21} = a_{31} = a_{41} = a_{51} = a_{32} = a_{42} = a_{52} = a_{43} = a_{53} = a_{54} = 0$ .

Fra  $s_1 \leq r_1$  følger det at  $a_{11} = s_1$  og  $a_{21} = a_{31} = a_{41} = a_{51} = 0$ . Videre kan vi ha ulike tilfeller.

1. Hvis  $r_1 \geq s_1 + s_2 + s_3 + s_4$ , gir regelen  $a_{12} = s_1$ ,  $a_{13} = s_3$ ,  $a_{14} = s_4$ , mens resten av elementene i kolonne to, tre og fire er null.

2. Hvis  $r_1 \geq s_1 + s_2 + s_3$ , mens  $r_1 - s_1 - s_2 - s_3 < s_4$ , gir første ulikhet at  $a_{12} = s_1$ ,  $a_{13} = s_3$  og at resten av elementene i kolonne to og tre er null. Videre gir andre ulikhet at  $a_{14} = r_1 - s_1 - s_2 - s_3$  og  $a_{15} = 0$ .

(a) Hvis  $r_2 \geq s_4 + s_3 + s_2 + s_1 - r_1$ , gir regelen  $a_{24} = s_4 + s_3 + s_2 + s_1 - r_1$  og  $a_{34} = a_{44} = a_{54} = 0$ .

(b) Hvis  $r_2 < s_4 + s_3 + s_2 + s_1 - r_1$ , gir regelen  $a_{24} = r_2$  og  $a_{25} = 0$ .

i. Hvis  $r_3 \geq s_4 + s_3 + s_2 + s_1 - r_1 - r_2$ , gir regelen  $a_{34} = s_4 + s_3 + s_2 + s_1 - r_1 - r_2$  og  $a_{44} = a_{54} = 0$ .

ii. Hvis  $r_3 < s_4 + s_3 + s_2 + s_1 - r_1 - r_2$ , gir regelen  $a_{34} = r_3$  og  $a_{35} = 0$ . Fordi  $s_4 + s_3 + s_2 + s_1 - r_1 - r_2 - r_3 \leq r_4$ , følger det at  $a_{44} = s_4 + s_3 + s_2 + s_1 - r_1 - r_2 - r_3$  og  $a_{54} = 0$ .

3. Hvis  $r_1 \geq s_1 + s_2$ , mens  $r_1 - s_1 - s_2 < s_3$ , gir første ulikhet  $a_{12} = s_2$  og at resten av elementene i kolonne to er null. Videre gir andre ulikhet at  $a_{13} = r_1 - s_1 - s_2$  og at resten av elementene i første rad er null.

(a) Hvis  $r_2 \geq s_3 + s_2 + s_1 - r_1$ , gir regelen  $a_{23} = s_3 + s_2 + s_1 - r_1$  og  $a_{33} = a_{43} = a_{53} = 0$ .

i. Hvis  $r_2 + r_1 - s_1 - s_2 - s_3 \geq s_4$ , gir regelen  $a_{24} = s_4$  og  $a_{34} = a_{44} = a_{54} = 0$ .

ii. Hvis  $r_2 + r_1 - s_1 - s_2 - s_3 < s_4$ , gir regelen  $a_{24} = r_2 + r_1 - s_1 - s_2 - s_3$  og  $a_{25} = 0$ .

A. Hvis  $r_3 \geq s_4 + s_3 + s_2 + s_1 - r_1 - r_2$ , gir regelen  $a_{34} = s_4 + s_3 + s_2 + s_1 - r_1 - r_2$  og  $a_{44} = a_{54} = 0$ .

B. Hvis  $r_3 < s_4 + s_3 + s_2 + s_1 - r_1 - r_2$ , gir regelen  $a_{34} = r_3$  og  $a_{35} = 0$ . Fordi  $s_4 + s_3 + s_2 + s_1 - r_1 - r_2 - r_3 \leq r_4$ , følger det videre at  $a_{44} = s_4 + s_3 + s_2 + s_1 - r_1 - r_2 - r_3$  og  $a_{54} = 0$ .

(b) Hvis  $r_2 < s_3 + s_2 + s_1 - r_1$ , gir regelen  $a_{23} = r_2$  og  $a_{24} = a_{25} = 0$ . Fordi  $s_3 + s_2 + s_1 - r_1 - r_2 \leq r_3$ , følger det videre at  $a_{33} = s_3 + s_2 + s_1 - r_1 - r_2$  og  $a_{43} = a_{53} = 0$ .

i. Hvis  $r_3 + r_2 + r_1 - s_1 - s_2 - s_3 \geq s_4$ , gir regelen at  $a_{34} = s_4$  og  $a_{44} = a_{54} = 0$ .

ii. Hvis  $r_3 + r_2 + r_1 - s_1 - s_2 - s_3 < s_4$ , gir regelen at  $a_{34} = r_3 + r_2 + r_1 - s_1 - s_2 - s_3$  og  $a_{35} = 0$ . Fordi  $s_4 + s_3 + s_2 + s_1 - r_1 - r_2 - r_3 \leq r_4$ , følger det videre at  $a_{44} = s_4 + s_3 + s_2 + s_1 - r_1 - r_2 - r_3$  og  $a_{54} = 0$ .

4. Hvis  $r_1 - s_1 < s_2$ , gir regelen at  $a_{12} = r_1 - s_1$  og at resten av elementene i første rad er null. Fordi  $s_2 + s_1 - r_1 \leq r_2$ , følger det videre at  $a_{22} = s_2 + s_1 - r_1$  og  $a_{32} = a_{42} = a_{52} = 0$ .

(a) Hvis  $r_2 + r_1 - s_1 - s_2 \geq s_3$ , gir regelen  $a_{23} = s_3$  og  $a_{33} = a_{43} = a_{53} = 0$ .

i. Hvis  $r_2 + r_1 - s_1 - s_2 - s_3 \geq s_4$ , gir regelen  $a_{24} = s_4$  og  $a_{34} = a_{44} = a_{54} = 0$ .

ii. Hvis  $r_2 + r_1 - s_1 - s_2 - s_3 < s_4$ , gir regelen  $a_{24} = r_2 + r_1 - s_1 - s_2 - s_3$  og  $a_{25} = 0$ .

A. Hvis  $r_3 \geq s_4 + s_3 + s_3 + s_1 - r_1 - r_2$ , gir regelen  $a_{34} = s_4 + s_3 + s_3 + s_1 - r_1 - r_2$  og  $a_{44} = a_{54} = 0$ .

B. Hvis  $r_3 < s_4 + s_3 + s_3 + s_1 - r_1 - r_2$ , gir regelen  $a_{34} = r_3$  og  $a_{35} = 0$ . Fordi  $s_4 + s_3 + s_2 + s_1 - r_1 - r_2 - r_3 \leq r_4$ , følger det videre at  $a_{44} = s_4 + s_3 + s_2 + s_1 - r_1 - r_2 - r_3$  og  $a_{54} = 0$ .

(b) Hvis  $r_2 + r_1 - s_1 - s_2 < s_3$ , gir regelen  $a_{23} = r_2 + r_1 - s_1 - s_2$  og  $a_{24} = a_{25} = 0$ . Fordi  $s_3 + s_2 + s_1 - r_1 - r_2 \leq r_3$ , følger det videre at  $a_{33} = s_3 + s_2 + s_1 - r_1 - r_2$  og  $a_{43} = a_{53} = 0$ .

i. Hvis  $r_3 + r_2 + r_1 - s_1 - s_2 - s_3 \geq s_4$ , gir regelen  $a_{34} = s_4$  og  $a_{44} = a_{54} = 0$ .

ii. Hvis  $r_3 + r_2 + r_1 - s_1 - s_2 - s_3 < s_4$ , gir regelen  $a_{34} = r_3 + r_2 + r_1 - s_1 - s_2 - s_3$  og  $a_{35} = 0$ . Fordi  $s_4 + s_3 + s_1 + s_1 - r_1 - r_2 - r_3 \leq r_4$ , følger det videre at  $a_{44} = s_4 + s_3 + s_2 + s_1 - r_1 - r_2 - r_3$  og  $a_{54} = 0$ .

### Induksjonsbevis tilhørende delkapittel 7.3

Viser ved induksjon at dersom to vektorer  $R = (r_1, \dots, r_n)$  og  $S = (s_1, \dots, s_n)$  oppfyller (7.3) og (7.4), så produserer Nordvest-hjørne regelen en matrise  $A \in \mathcal{N}_W(R, S)$ , der  $W$  er trappematriksen med  $v_i = i$  for  $i = 1, \dots, n$  og  $h_i = i + 1$  for  $i = 1, \dots, n - 1$ ,  $h_n = n$ .

*Bevis.* Anta at  $R$  og  $S$  oppfyller (7.3) og (7.4). Ved å bruke Nordvest-hjørne regelen til å fullføre elementene til og med kolonne 3 får vi delmatriksen:

$$\begin{bmatrix} s_1 & r_1 - s_1 & 0 & 0 & \dots & 0 \\ 0 & s_2 + s_1 - r_1 & r_2 + r_1 - s_1 - s_2 & 0 & \dots & 0 \\ 0 & 0 & s_3 + s_2 + s_1 - r_1 - r_2 & & & \\ 0 & 0 & 0 & & & \\ \vdots & \vdots & \vdots & & & \\ 0 & 0 & 0 & & & \end{bmatrix}$$

Anta videre at det å bruke Nordvest-hjørne regelen til å fullføre elementene til og med rad  $p - 1$  og kolonne  $p$ , gir samme mønster. Det vil si at for  $k = 2, \dots, p$ , så er:

### A.3. Nordvest-Hjørne regelen produserer tridiagonale transportmatriser

- $a_{k-1,k} = r_{k-1} + \dots + r_1 - s_1 - \dots - s_{k-1}$ , mens  $a_{k-1,j} = 0$  for  $j > k$ .
- $a_{kk} = s_k + \dots + s_1 - r_1 - \dots - r_{k-1}$ , mens  $a_{jk} = 0$  for  $j > k$ .

Gitt denne antakelsen skal vi vise at Nordvest-hjørne regelen også gir rad  $p$  og kolonne  $p+1$  på samme form. Neste element som skal bestemmes er  $a_{p,p+1}$ . Fra antakelsen følger det at alle elementer i kolonne  $p+1$  som står over  $a_{p,p+1}$  er lik null og at alle elementer i rad  $p$  som står til venstre for  $a_{p,p}$  er lik null. Vi sammenligner derfor  $s_{p+1}$  med følgende oppdaterte radsum for rad  $p$ .

$$r'_p = r_p - a_{pp} = r_p - (s_p + \dots + s_1 - r_1 - \dots - r_{p-1}) = r_p + \dots + r_1 - s_1 - \dots - s_p.$$

Fra (7.4) har vi at  $r'_p \leq s_{p+1}$ . Nordvest-hjørne regelen gir dermed  $a_{p,p+1} = r'_p$  og at  $a_{pj} = 0$  for  $j > p+1$ . Neste element som skal bestemmes er derfor  $a_{p+1,p+1}$ . Fra antakelsen følger det at alle elementer i rad  $p+1$  som står til venstre for  $a_{p+1,p+1}$  er lik null. Vi sammenligner derfor  $r_{p+1}$  med følgende oppdaterte kolonnesum for kolonne  $p+1$ :

$$s'_{p+1} = s_{p+1} - a_{p,p+1} = s_{p+1} - (r_p + \dots + r_1 - s_1 - \dots - s_p) = s_{p+1} + \dots + s_1 - r_1 - \dots - r_p.$$

Fra (7.3) har vi at  $s'_{p+1} \leq r_{p+1}$ . Nordvest-hjørne regelen gir dermed  $a_{p+1,p+1} = s'_{p+1}$ , mens  $a_{j,p+1} = 0$  for  $j > p+1$ . ■

### A.3 Nordvest-Hjørne regelen produserer tridiagonale transportmatriser

Anta at to vektorer  $R$  og  $S$  oppfyller (7.5) for  $n = 3$ . For å vise at Nordvest-Hjørne regelen produserer en tridiagonal  $3 \times 3$  matrise  $A = [a_{ij}]$  med radsumvektor  $R$  og kolonnesumvektor  $S$ , må vi vise at  $a_{13} = a_{31} = 0$ .

1. Hvis  $r_1 \leq s_1$ , gir regelen  $a_{11} = r_1$  og  $a_{12} = a_{13} = 0$ . Fordi  $s_1 - r_1 \leq r_2$  følger det videre at  $a_{21} = s_1 - r_1$  og  $a_{31} = 0$ .
2. Hvis  $r_1 > s_1$ , gir regelen  $a_{11} = s_1$  og  $a_{21} = a_{31} = 0$ . Fordi  $r_1 - s_1 \leq s_2$  følger det videre at  $a_{12} = r_1 - s_1$  og  $a_{13} = 0$ .

Anta at to vektorer  $R$  og  $S$  oppfyller (7.5) for  $n = 4$ . For å vise at Nordvest-Hjørne regelen produserer en tridiagonal  $4 \times 4$  matrise  $A = [a_{ij}]$  med radsumvektor  $R$  og kolonnesumvektor  $S$ , må vi vise at  $a_{13} = a_{14} = a_{24} = a_{31} = a_{41} = a_{42} = 0$ .

1. Hvis  $r_1 \leq s_1$ , gir regelen  $a_{11} = r_1$  og  $a_{12} = a_{13} = a_{14} = 0$ . Fordi  $s_1 - r_1 \leq r_2$  følger det videre at  $a_{21} = s_1 - r_1$  og  $a_{31} = a_{41} = 0$ .
  - (a) Hvis  $r_2 + r_1 - s_1 \leq s_2$ , gir regelen  $a_{22} = r_2 + r_1 - s_1$  og  $a_{23} = a_{24} = 0$ . Fordi  $s_1 + s_2 - r_1 - r_2 \leq r_3$  følger det videre at  $a_{32} = s_2 + s_1 - r_1 - r_2$  og  $a_{42} = 0$ .
  - (b) Hvis  $r_2 + r_1 - s_1 > s_2$ , gir regelen  $a_{22} = s_2$  og  $a_{32} = a_{42} = 0$ . Fordi  $r_2 + r_1 - s_1 - s_2 \leq s_3$  følger det videre at  $a_{23} = r_2 + r_1 - s_1 - s_2$  og  $a_{24} = 0$ .
2. Hvis  $r_1 > s_1$ , gir regelen  $a_{11} = s_1$  og  $a_{21} = a_{31} = a_{41} = 0$ . Fordi  $r_1 - s_1 \leq s_2$  følger det videre at  $a_{12} = r_1 - s_1$  og  $a_{13} = a_{14} = 0$ .
  - (a) Hvis  $r_2 \leq s_2 + s_1 - r_1$ , gir regelen  $a_{22} = r_2$  og  $a_{23} = a_{24} = 0$ . Fordi  $s_2 + s_1 - r_1 - r_2 \leq r_3$  følger det videre at  $a_{32} = s_2 + s_1 - r_1 - r_2$  og  $a_{42} = 0$ .

- (b) Hvis  $r_2 > s_2 + s_1 - r_1$ , gir regelen  $a_{22} = s_2 + s_1 - r_1$  og  $a_{32} = a_{42} = 0$ . Fordi  $r_1 + r_2 - s_1 - s_2 \leq s_3$  følger det videre at  $a_{23} = r_2 + r_1 - s_1 - s_2$  og  $a_{24} = 0$ .

Anta at to vektorer  $R$  og  $S$  oppfyller (7.5) for  $n = 5$ . For å vise at Nordvest-Hjørne regelen produserer en tridiagonal  $5 \times 5$  matrise  $A = [a_{ij}]$  med radsumvektor  $R$  og kolonnesumvektor  $S$ , må vi vise at  $a_{13} = a_{14} = a_{15} = a_{24} = a_{25} = a_{31} = a_{35} = a_{41} = a_{42} = a_{51} = a_{52} = a_{53} = 0$ .

1. Hvis  $r_1 \leq s_2$ , gir regelen  $a_{11} = r_1$  og  $a_{12} = a_{13} = a_{14} = a_{15} = 0$ . Fordi  $s_1 - r_1 \leq r_2$  følger det videre at  $a_{21} = s_1 - r_1$  og  $a_{31} = a_{41} = a_{51} = 0$ .

- (a) Hvis  $r_2 + r_1 - s_1 \leq s_2$ , gir regelen  $a_{22} = r_2 + r_1 - s_1$  og  $a_{23} = a_{24} = a_{25} = 0$ . Fordi  $s_2 + s_1 - r_1 - r_2 \leq r_3$  følger det videre at  $a_{32} = s_2 + s_1 - r_1 - r_2$  og  $a_{42} = a_{52} = 0$ .

i. Hvis  $r_3 + r_2 + r_1 - s_2 - s_1 \leq s_3$ , gir regelen  $a_{33} = r_3 + r_2 + r_1 - s_2 - s_1$  og  $a_{34} = a_{35} = 0$ . Fordi  $s_3 + s_2 + s_1 - r_1 - r_2 - r_3 \leq r_4$  følger det videre at  $a_{43} = s_3 + s_2 + s_1 - r_1 - r_2 - r_3$  og  $a_{53} = 0$ .

ii. Hvis  $r_3 + r_2 + r_1 - s_2 - s_1 > s_3$ , gir regelen  $a_{33} = s_3$  og  $a_{43} = a_{53} = 0$ . Fordi  $r_3 + r_2 + r_1 - s_1 - s_2 - s_3 \leq s_4$  følger det videre at  $a_{34} = r_3 + r_2 + r_1 - s_1 - s_2 - s_3$  og  $a_{35} = 0$ .

- (b) Hvis  $r_2 + r_1 - s_1 > s_2$ , gir regelen  $a_{22} = s_2$  og  $a_{32} = a_{42} = a_{52} = 0$ . Fordi  $r_1 + r_2 - s_1 - s_2 \leq s_3$  følger det videre at  $a_{23} = r_2 + r_1 - s_1 - s_2$  og  $a_{24} = a_{25} = 0$ .

i. Hvis  $r_3 \leq s_3 + s_2 + s_1 - r_1 - r_2$ , gir regelen  $a_{33} = r_3$  og  $a_{34} = a_{35} = 0$ . Fordi  $s_3 + s_2 + s_1 - r_1 - r_2 - r_3 \leq r_4$  følger det videre at  $a_{43} = s_3 + s_2 + s_1 - r_1 - r_2 - r_3$  og  $a_{53} = 0$ .

ii. Hvis  $r_3 > s_3 + s_2 + s_1 - r_1 - r_2$ , gir regelen  $a_{33} = s_3 + s_2 + s_1 - r_1 - r_2$  og  $a_{43} = a_{53} = 0$ . Fordi  $r_3 + r_2 + r_1 - s_1 - s_2 - s_3 \leq s_4$  følger det videre at  $a_{34} = r_3 + r_2 + r_1 - s_1 - s_2 - s_3$  og  $a_{35} = 0$ .

2. Hvis  $r_1 > s_1$ , gir regelen  $a_{11} = s_1$  og  $a_{21} = a_{31} = a_{41} = a_{51} = 0$ . Fordi  $r_1 - s_1 \leq s_2$  følger det videre at  $a_{12} = r_1 - s_1$  og  $a_{13} = a_{14} = a_{15} = 0$ .

- (a) Hvis  $r_2 \leq s_2 + s_1 - r_1$ , gir regelen  $a_{22} = r_2$  og  $a_{23} = a_{24} = a_{25} = 0$ . Fordi  $s_2 + s_1 - r_1 - r_2 \leq r_3$ , følger det videre at  $a_{32} = s_2 + s_1 - r_1 - r_2$  og  $a_{42} = a_{52} = 0$ .

i. Hvis  $r_3 + r_2 + r_1 - s_1 - s_2 \leq s_3$ , gir regelen  $a_{33} = r_3 + r_2 + r_1 - s_1 - s_2$  og  $a_{34} = a_{35} = 0$ . Fordi  $s_3 + s_2 + s_1 - r_1 - r_2 - r_3 \leq r_4$  følger det videre at  $a_{43} = s_3 + s_2 + s_1 - r_1 - r_2 - r_3$  og  $a_{53} = 0$ .

ii. Hvis  $r_3 + r_2 + r_1 - s_1 - s_2 > s_3$ , gir regelen  $a_{33} = s_3$  og  $a_{43} = a_{53} = 0$ . Fordi  $r_3 + r_2 + r_1 - s_1 - s_2 - s_3 \leq s_4$  følger det videre at  $a_{34} = r_3 + r_2 + r_1 - s_1 - s_2 - s_3$  og  $a_{35} = 0$ .

- (b) Hvis  $r_2 > s_2 + s_1 - r_1$ , gir regelen  $a_{22} = s_2 + s_1 - r_1$  og  $a_{32} = a_{42} = a_{52} = 0$ . Fordi  $r_2 + r_1 - s_1 - s_2 \leq s_3$ , følger det videre at  $a_{23} = r_2 + r_1 - s_1 - s_2$  og  $a_{24} = a_{25} = 0$ .

i. Hvis  $r_3 \leq s_3 + s_2 + s_1 - r_1 - r_2$ , gir regelen  $a_{33} = r_3$  og  $a_{34} = a_{35} = 0$ . Fordi  $s_3 + s_2 + s_1 - r_1 - r_2 - r_3 \leq r_4$  følger det videre at  $a_{43} = s_3 + s_2 + s_1 - r_1 - r_2 - r_3$  og  $a_{53} = 0$ .



### A.3. Nordvest-Hjørne regelen produserer tridiagonale transportmatriser

- ii. Hvis  $r_3 > s_3 + s_2 + s_1 - r_1 - r_2$ , gir regelen  $a_{33} = s_3 + s_2 + s_1 - r_1 - r_2$  og  $a_{43} = a_{53} = 0$ . Fordi  $r_3 + r_2 + r_1 - s_1 - s_2 - s_3 \leq s_4$ , følger det videre at  $a_{34} = r_3 + r_2 + r_1 - s_1 - s_2 - s_3$  og  $a_{35} = 0$ .



## Tillegg B

# MATLAB-kode tilhørende kapittel 7.5

### B.1 Avgjør om det finnes en nedre triangulær transportmatrise

*% Funksjonen tar to vektorer R og S som argumenter og avgjør om det finnes  
% en nedre triangulær matrise med radsumvektor R og kolonnesumvektor S.  
% Funksjonen returnerer 1 hvis det finnes, og 0 om ikke.*

```
function a=nedre(R,S)
for k=1:1:length(R)-1
    if sum(R(1:k))>sum(S(1:k))
        a=0;
        return
    end
end
a=1;
fprintf(['En nedre triangulær %d x %d matrise med de gitte linjesummene' ...
        ' er matrisen:\n'],length(R),length(R))
end
```

### B.2 Avgjør om det finnes en øvre triangulær transportmatrise

*% Funksjonen tar to vektorer R og S som argumenter og avgjør om det finnes  
% en øvre triangulær matrise med radsumvektor R og kolonnesumvektor S.  
% Funksjonen returnerer 1 hvis det finnes, og 0 om ikke.*

```
function a=ovre(R,S)
for k=1:1:length(R)-1
    if sum(R(1:k))<sum(S(1:k))
        a=0;
        return
    end
end
a=1;
fprintf(['En %d x %d øvre triangulær matrise med de gitte linjesummene' ...
        ' er matrisen:\n'],length(R),length(R))
end
```

### B.3 Avgjør om det finnes en tridiagonal transportmatrise

*% Funksjonen tar to vektorer R og S som argumenter og avgjør om det finnes  
% en tridiagonal matrise med radsumvektor R og kolonnesumvektor S.  
% Funksjonen returnerer 1 hvis det finnes, og 0 om ikke.*

```
function a=tri(R,S)
for k=1:1:length(R)-2
    if sum(R(1:k))>sum(S(1:k+1))
        a=0;
        return
    end
end
for k=1:1:length(R)-2
    if sum(S(1:k))>sum(R(1:k+1))
        a=0;
        return
    end
end
a=1;
fprintf(['En tridiagonal %d x %d matrise med de gitte linjesummene' ...
        ' er matrisen:\n'],length(R),length(R))
end
```

## B.4 Lager nedre triangulære, øvre triangulære og tridiagonale transportmatriser

```

% Funksjonen tar to vektorer R og S som argumenter.
% Funksjonen sjekker først om R og S har samme lengde, og deretter om
% summen av elementene i R er lik summen av elementene i S.
% Dersom begge kravene oppfylles sjekker funksjonen om det eksisterer
% en nedre triangulær, øvre triangulær eller tridiagonal matrise
% med radsumvektor R og kolonnesumvektor S. Dersom det gjør det,
% løser funksjonen et tillatt strøm problem definert på nettverket som
% korresponderer med matrisen med den bestemte strukturen og
% linjesumvektorer $R$ og $S$. Funksjonen returnerer en
% figur av det transformerte nettverket med den beregnede maksimum
% st-strømmen, og den korresponderende matrisen.

% Notasjon i koden:
% n=antall rader=antall kolonner
% Node s betegnes med "s".
% Noden som korresponderer med rad i betegnes med "i"
% Noden som korresponderer med kolonne i betegnes med "n+i"
% Node t betegnes med "2*n+t"

function M =makematrix(R,S)
m=length(R);
n=length(S);
if m ~=n
    fprintf('R og S har forskjellig lengde\n');
    return
end
if sum(R) ~= sum(S)
    fprintf(['Summen av elementene i R er forskjellig fra summen av' ...
            ' elementene i S\n']);
    return
end
%Forbereder vektorer som skal inneholde startnoder, endenoder og kapasiteter.
% Det er n + n*n + n kanter i nettverket.
ut=zeros(1,n+(n*n)+n);
inn=zeros(1,n+(n*n)+n);
kap=zeros(1,n+(n*n)+n);
kap1=zeros(1,n+(n*n)+n);
% Det skal gå en kant fra node s til radnode i med kapasitet r_i.
for i=1:1:n
    ut(i)=1;
    inn(i)=i+1;
    kap(i)=R(i);
end
% Det skal gå en kant fra kolonnenode j til node t med kapasitet s_j.
for j=1:1:n
    ut(n+(n*n)+j)=n+1+j;

```

Tillegg B. MATLAB-kode tilhørende kapittel 7.5

```

    inn(n+(n*n)+j)=n+n+2;
    kap(n+(n*n)+j)=S(j);
end
% Det skal gå en kant fra hver radnode til hver kolonnenode.
for i=1:1:n
    ut(n+1+(i-1)*n:n+n*i)=i+1;
    for j=1:1:n
        inn(n+j+(i-1)*n)=n+1+j;
    end
end
for i=1:1:n
    for j=1:1:n
        kap1(n+j+(i-1)*n)=min(R(i),S(j));
    end
end
if nedre(R,S)==1
% Kanter som går mellom radnode i og kolonnenode j>=i, tilordnes kapasitet
% lik minimumet r_i og s_j. Resten av kantene bevarer kapasitet 0.
    for i=1:1:n
        for k=(n+1)*i-(i-1):(n+1)*i
            kap(k)=kap1(k);
        end
    end
elseif ovre(R,S)==1
% Kanter som går mellom radnode i og kolonnenode j<=i tilordnes kapasitet
% lik minimumet av r_i og s_j. Resten av kantene bevarer kapasitet 0.
    for i=1:1:n
        for k=(n+1)*i:1:(n+1)*i+(n-i)
            kap(k)=kap1(k);
        end
    end
elseif tri(R,S)==1
% For radnode i=2,...,n-1:
% Kanter som går mellom radnode i og kolonnenode j=i, j=i-1 og j=i+1
% tilordnes kapasitet lik minimumet r_i og s_j.
% Kanten som går mellom radnode 1 og kolonnenode 1 og kanten mellom
% radnode 1 og kolonnenode 2 tilordnes kapasitet lik henholdsvis
% minimumet av r_1 og s_1 og minimumet av r_1 og s_2.
% Tilsvarende tilordnes kanten som går mellom radnode n og
% kolonnenode n og kanten mellom radnode n og
% kolonnenode n-1 kapasitet lik minimumet av henholdsvis r_n og s_n og
% r_n og s_{n-1}. Resten av kantene bevarer kapasitet 0.
    for i=2:1:n-1
        for k=i*(n+1)-1:1:i*(n+1)+1
            kap(k)=kap1(k);
        end
    end
    for i=[n+1,n+2,n*(n+1)-1,n*(n+1)]
        kap(i)=kap1(i);
    end
end

```

#### B.4. Lager nedre triangulære, øvre triangulære og tridiagonale transportmatriser

```

    end
else
    fprintf(['Det finnes ingen nedre triangulær, ' ...
           'øvre triangulær eller tridiagonal ikke-negativ ' ...
           '%d x %d -matrise med de gitte linjesummene.\n'], n,n)
    return
end
% Anvender den innebygde MATLAB-funksjoner "digraph" som tar
% vektorer med startnoder, endenoder og kapasiteter som argumenter
% og returnerer en tabell med denne dataen. Videre anvendes
% MATLAB-funksjonen "plot" for å plote dataen i tabellen som et nettverk.
% Deretter benyttes MATLAB-funksjonen "maxflow" for å bestemme en maksimum
% st-strøm i nettverket ved hjelp av Ford-Folkerson algoritmen.
% Til slutt benyttes MATLAB-funksjonen "labeledge" for å markere
% maksimumstrømmen i nettverket.
G = digraph(ut,inn,kap);
H = plot(G, 'EdgeLabel', G.Edges.Weight);
[~,GF] = maxflow(G,1,n+n+2, 'augmentpath');
H.EdgeLabel = {};
highlight(H,GF, 'EdgeColor', 'r', 'LineWidth', 2);
st = GF.Edges.EndNodes;
labeledge(H,st(:,1),st(:,2),GF.Edges.Weight);
camroll(90)
% Henter ut strømløsningen og bestemmer hvilke elementer som
% korresponderer med kanter fra node s eller til node t. Disse elementene
% skal ikke med i nettverkets korresponderende matrise.
A=GF.Edges.EndNodes;
F=GF.Edges.Weight;
l=length(F);
b=0;
c=0;
for i=1:1:n
    if A(i,1)==1
        b=b+1;
    end
end
for i=1-n+1:1:l
    if A(i,2)==n+n+2
        c=c+1;
    end
end
% Konstruerer matrisen som korresponderer med strømløsningen.
M=zeros(n,n);
for i=b+1:1:l-c
    M(A(i,1)-1,A(i,2)-(n+1))=F(i);
end
end

```





## Tillegg C

# MATLAB-kode tilhørende kapittel 8.5

### C.1 Lager (0,1)-matriser med to rader og gitte linjesummer

```
% Funksjonen tar to vektorer R og S som argumenter.
% Funksjonen sjekker først om R og S oppfyller ulikheten i Teorem 8.3.1.
% Dersom de gjør det returneres en (0,1)-matrise med to rader,
% radsumvektor R og kolonnesumvektor S. Om ikke returnerer funksjonen
% en kommentar om at R og S ikke oppfyller ulikheten.
function M = matrisefunksjon2(R,S)
    m = length(R);
    n=length(S);
    p=0;
    q=0;
    % Bestemmer antall kolonner med kolonnesum lik 2 og 0.
    for i=1:1:n
        if S(i)==m
            p=p+1;
        elseif S(i)==0
            q=q+1;
        end
    end
    % Sjekker om R oppfyller R(1) <= n-q.
    if R(1)>n-q
        fprintf('R(1)> n-q\n');
        M=zeros(m,n);
        return
    end
    % Oppdaterer elementene i R, når vi ser bort i fra kolonner
    % som skal ha kolonnesum lik 2.
    for i=1:1:m
        R(i)=R(i)-p;
    end
    n=n-p-q;
    M=zeros(m,n);
    % Erstatter nullere med enere på en slik måte at den resulterende
    % matrisen får radsummer lik elementene i den oppdaterte radsumvektoren,
    % og kolonnesummer lik elementene i S uten de p første og q siste elementene.
```

```

for i=1:1:R(1)
    M(1,i)=1;
end
for i=R(1)+1:1:n
    M(2,i)=1;
end
a=[1;1];
b=[0;0];
% Legger til p kolonner til venstre med kolonnesum lik 1
% og q kolonner til høyre med kolonnesum lik 0.
for i=1:1:p
    M=[a M];
end
for i=1:1:q
    M=[M b];
end
end
end

```

## C.2 Lager (0,1)-matriser med tre rader og gitte linjesummer

*% Funksjonen tar to vektorer R og S som argumenter.  
 % Funksjonen sjekker først om R og S oppfyller ulikhetene i Teorem 8.3.1.  
 % Dersom de gjør det returneres en (0,1)-matrise med tre rader, radsumvektor  
 % R og kolonnesumvektor S. Om ikke returnerer funksjonen en kommentar om  
 % hvilken av ulikhetene R og S ikke oppfyller.*

```

function M=matrisefunksjon3(R,S)
    m = length(R);
    n=length(S);
    p=0;
    q=0;
    l=0;
    % Bestemmer antall kolonner med kolonnesum lik 3, 2 og 0.
    for i=1:1:n
        if S(i)==m
            p=p+1;
        elseif S(i)==0
            q=q+1;
        elseif S(i)==2
            l=l+1;
        end
    end
    end
    % Sjekker om R og S oppfyller ulikhetene i Teorem 8.3.1.
    if R(1)>n-q
        fprintf('R(1)> n-q\n');
        M=zeros(m,n);
        return
    elseif R(m)<p
        fprintf('R(%d)< p\n',m);
    end
end

```

## C.2. Lager (0,1)-matriser med tre rader og gitte linjesummer

```

    M=zeros(m,n);
    return
end
% Oppdaterer elementene i R, når vi ser bort i fra
% kolonner med kolonnesum 3.
for i=1:1:m
    R(i)=R(i)-p;
end
n=n-p-q;
M=zeros(m,n);
% Erstatte nullere med enere på en slik måte at den resulterende
% matrisen får radsummer lik elementene i den oppdaterte radsumvektoren,
% og kolonnesummer lik elementene i S uten de p første og q siste elementene.
% Underveis i koden refereres det til hvilket av tilfellene beskrevet i
% delkapittel 7.3 kodesekvensen tar for seg.

if R(1)>=1 && R(2)>=1 % "Tilfelle 1a)"
    for i=1:1:R(1)
        M(1,i)=1;
    end
    for i=1:1:1
        M(2,i)=1;
    end
    if R(2)>1
        for i=R(1)+1:1:R(1)+R(2)-1
            M(2,i)=1;
        end
    end
    for i=R(1)+(R(2)-1)+1:1:n
        M(3,i)=1;
    end
elseif R(1)>=1 && R(2)<1 % "Tilfelle 1b)"
    for i=1:1:R(1)
        M(1,i)=1;
    end
    for i=1:1:R(2)
        M(2,i)=1;
    end
    for i=R(2)+1:1:1
        M(3,i)=1;
    end
    for i=R(1)+1:1:n
        M(3,i)=1;
    end
elseif R(1)<1 % "Tilfelle 2"
    for i=1:1:R(1)
        M(1,i)=1;
    end
    for i=R(1)+1:1:1

```

```

        M(2,i)=1;
    end
    for i=1:1:R(2)-(1-R(1))
        M(2,i)=1;
    end
    for i=R(2)-(1-R(1))+1:1:n
        M(3,i)=1;
    end

end
a=[1;1;1];
b=[0;0;0];
% Legger til p kolonner til venstre med kolonnesum lik 1
% og q kolonner til høyre med kolonnesum lik 0.
for i=1:1:p
    M=[a M];
end
for i=1:1:q
    M=[M b];
end
end
end

```

### C.3 Lager (0,1)-matriser med fire rader og gitte linjesummer

```

% Funksjonen tar to vektorer R og S som argumenter og sjekker om R og S
% oppfyller ulikhetene i Teorem 8.4.1. Dersom de gjør det
% returneres en (0,1)-matrise med fire rader og linjesummer lik R og S.
% Om ikke returnerer funksjonen en kommentar om hvilken av ulikhetene
% R og S ikke oppfyller.

```

```

function M= matrisefunksjon4(R,S)
    m = length(R);
    n=length(S);
    p=0;
    q=0;
    l=0;
    k=0;
    % Bestemmer antall kolonner med sum lik 4,3,2 og 0.
    for i=1:1:n
        if S(i)==m
            p=p+1;
        elseif S(i)==0
            q=q+1;
        elseif S(i)==2
            l=l+1;
        elseif S(i)==3
            k=k+1;
        end
    end
    end
    % Sjekker om R og S oppfyller ulikhetene i Teorem 8.4.1.

```

```

if R(1)>n-q
    fprintf('R(1)> n-q\n');
    M=zeros(m,n);
    return
elseif R(m)<p
    fprintf('R(%d)< p\n',m);
    M=zeros(m,n);
    return
elseif R(1)+R(2)>n-q+p+k+1
    fprintf('R(1)+R(2)>n-q+p+k+1\n');
    M=zeros(m,n);
    return
end
% Oppdaterer elementene i R, når vi ser bort i fra
% kolonner med kolonnesum 4.
for i=1:1:m
    R(i)=R(i)-p;
end
n=n-p-q;
M=zeros(m,n);
% Erstatter nullere med enere på en slik måte at den resulterende
% matrisen får radsummer lik elementene i den oppdaterte radsumvektoren,
% og kolonnesummer lik elementene i S uten de p første og q siste elementene.
% Underveis i koden refereres det til hvilket av tilfellene beskrevet i
% delkapittel 7.4 kodesekvensen tar for seg.
if R(1)>=(k+1) && R(2)>=(k+1) && R(3)>=k % "Tilfelle 1a"
    for i=1:1:R(1)
        M(1,i)=1;
    end
    for i=1:1:k+1
        M(2,i)=1;
    end
    if R(2)>k+1
        for i=R(1)+1:1:R(1)+R(2)-(k+1)
            M(2,i)=1;
        end
    end
    for i=1:1:k
        M(3,i)=1;
    end
    if R(3)>k
        for i=R(1)+R(2)-(k+1)+1:1:R(1)+R(2)-(k+1)+R(3)-k
            M(3,i)=1;
        end
    end
    for i=R(1)+R(2)-(k+1)+R(3)-k+1:1:n
        M(4,i)=1;
    end
elseif R(1)>=(k+1) && R(2)>=(k+1) && R(3)<k % "Tilfelle 1b"

```

```

    for i=1:1:R(1)
        M(1,i)=1;
    end
    for i=1:1:k+1
        M(2,i)=1;
    end
    if R(2)>k+1
        for i=R(1)+1:1:R(1)+R(2)-(k+1)
            M(2,i)=1;
        end
    end
    for i=1:1:R(3)
        M(3,i)=1;
    end
    for i=R(3)+1:1:k
        M(4,i)=1;
    end
    for i=R(1)+R(2)-(k+1)+1:1:n
        M(4,i)=1;
    end
elseif R(1)>=(k+1) && R(2)<k+1 % "Tilfelle 2"
    for i=1:1:R(1)
        M(1,i)=1;
    end
    for i=1:1:R(2)
        M(2,i)=1;
    end
    if R(3)>=(k+1)-R(2)+k % "2a)"
        for i=R(2)+1:1:k+1
            M(3,i)=1;
        end
        for i=1:1:k
            M(3,i)=1;
        end
        for i=R(1)+1:1:R(1)+R(2)+R(3)-(k+1)-k
            M(3,i)=1;
        end
        for i=R(1)+R(2)+R(3)-(k+1)-k+1:1:n
            M(4,i)=1;
        end
    end
elseif R(3)>=(k+1)-R(2) && R(3)<(k+1)-R(2)+k % "2b)"
    for i=R(2)+1:1:k+1
        M(3,i)=1;
    end
    for i=1:1:R(3)+R(2)-(k+1)
        M(3,i)=1;
    end
    for i=R(3)+R(2)-(k+1)+1:1:k

```

### C.3. Lager (0,1)-matriser med fire rader og gitte linjesummer

```

        M(4,i)=1;
    end
    for i=R(1)+1:1:n
        M(4,i)=1;
    end
elseif R(3)<k+1-R(2) % "2c)"
    for i=R(2)+1:1:R(2)+R(3)
        M(3,i)=1;
    end
    for i=R(2)+R(3)+1:1:k+1
        M(4,i)=1;
    end
    for i=1:1:k
        M(4,i)=1;
    end
    for i=R(1)+1:1:n
        M(4,i)=1;
    end
elseif R(1)<k+1 % "Tilfelle 3"
    for i=1:1:R(1)
        M(1,i)=1;
    end
    for i=R(1)+1:1:k+1
        M(2,i)=1;
    end
    for i=1:1:R(2)+R(1)-(k+1)
        M(2,i)=1;
    end
    if R(3)>=2*(k+1)-R(1)-R(2)+k % "3a)"
        for i=R(2)+R(1)-(k+1)+1:1:k+1
            M(3,i)=1;
        end
        for i=1:1:k
            M(3,i)=1;
        end
        for i=k+1+1:1:R(1)+R(2)+R(3)-(k+1)-k
            M(3,i)=1;
        end
        for i=R(1)+R(2)+R(3)-(k+1)-k+1:1:n
            M(4,i)=1;
        end
    elseif R(3)>=2*(k+1)-R(1)-R(2) && R(3)<2*(k+1)-R(1)-R(2)+k % "3b)"
        for i=R(2)+R(1)-(k+1)+1:1:k+1
            M(3,i)=1;
        end
        for i=1:1:R(1)+R(2)+R(3)-2*(k+1)
            M(3,i)=1;
        end
        for i=R(1)+R(2)+R(3)-2*(k+1)+1:k

```

```

        M(4,i)=1;
    end
    for i=(k+1)+1:1:n
        M(4,i)=1;
    end
elseif R(3)<2*(k+1)-R(1)-R(2) % "3c)"
    for i=R(2)+R(1)-(k+1)+1:1:R(2)+R(1)-(k+1)+R(3)
        M(3,i)=1;
    end
    for i=R(2)+R(1)-(k+1)+R(3)+1
        M(4,i)=1;
    end
    for i=1:1:k
        M(4,i)=1;
    end
    for i=(k+1)+1:1:n
        M(4,i)=1;
    end
end
end
a=[1;1;1;1];
b=[0;0;0;0];
% Legger til p kolonner til venstre med kolonnesum lik 1
% og q kolonner til høyre med kolonnesum lik 0.
for i=1:1:p
    M=[a M];
end
for i=1:1:q
    M=[M b];
end
end
end

```

#### C.4 Returnerer en (0,1)-matrise med to, tre eller fire rader og gitte linjesummer

```

% Funksjonen tar to vektorer R og S som argumenter.
% Funksjonen sjekker først om summen av elementene i R er lik summen
% av elementene i S. Dersom de er like bestemmes lengden av R.
% Hvis R har lengde lik 2, 3 eller 4 R og S oppfyller ulikhetene i
% henholdsvis Teorem 8.2.1., Teorem 8.3.1. eller Teorem 8.4.1. så
% returnerer funksjonen en (0,1)-matrise med radsumvektor R og
% kolonnesumvektor S.

function M=matrisefunksjon(R,S)
% Sjekker at summen av elementene i R er lik summen av elementene i S.
if sum(R)~= sum(S)
    fprintf(['Summen av elementene i R er forskjellig fra...' ...
            ' summen av elementene i S.\n']);
    return
end

```



C.4. Returnerer en (0,1)-matrise med to, tre eller fire rader og gitte linjesummer

```
end
m=length(R); % Bestemmer lengden til R.
if m==2
    M=matrisefunksjon2(R,S);
elseif m==3
    M=matrisefunksjon3(R,S);
elseif m==4
    M=matrisefunksjon4(R,S);
else
    fprintf('R har ikke riktig dimensjon.\n');
end
```