KAPITEL II

PARDANNELSE.

II.1 INDLEDNING

Begrebet "pardannelse" dækker over mange forskellige problem-typer. Disse problem-typer dukker op i flere forskellige sammenhænge og er af fundamental betydning i algoritmisk kombinatorik. Vi så således, at pardannelse spiller en vigtig rolle i ALGORITME POSTBUD og i ALGORITME HANDELSREJSENDE. Som et tredie typisk eksempel kan man tænke på en skoles skema-lægning, som jo består i at danne par af klasser og lærere i hver skematime.

Eksempel 2.1 Lad 5×5-matricen

$$
\begin{array}{c|c|c|c|c|c|}
 & B_1 & B_2 & B_3 & B_4 & B_5 \\
\hline
A_1 & 3 & 4 & 5 & 2 & 1 \\
\hline
A_2 & 4 & 5 & 5 & 3 & 1 \\
\hline
A_3 & 4 & 4 & 5 & 1 & 2 \\
\hline
A_4 & 4 & 2 & 4 & 2 & 2 \\
\hline
A_5 & 3 & 2 & 5 & 3 & 3 \\
\hline
\end{array}
$$

være givet. Problemet er at finde 5 *skæve* pladser, dvs. at ikke to er i samme række eller søjle, således at summen af tallene på de 5 pladser er mindst mulig. Dette problem er af en type som kaldes *job-tilordning*, idet de 5 rækker $A_1, A_2, \ldots, A_5$ i matricen kan svare til 5 ansøgere og de 5 søjler $B_1, B_2, \ldots, B_5$ kan svare til 5 jobs, som skal besættes. Tallet c_{ij} i række i og søjle j angiver så ansøger A_i's uegnethed til job B_j (jo større c_{ij} er, jo mere uegnet er ansøger A_i til job B_j). Vi ønsker at besætte de 5 jobs med de 5 ansøgere, således at den samlede "uegnethed" er mindst mulig.

Problem-typen kaldes også af og til for *vægtet to-delt pardannelse*. Ordene "to-delt pardannelse" går på, at vi skal danne par, hvor hvert par indeholder ét element af hver af de to typer $\{A_1 , A_2 , \ldots\}$ og $\{B_1 , B_2 , \ldots\}$. Ordet "vægtet" hentyder til, at de enkelte mulige par (A_i , B_i) har vægte c_{ij} (i *ikke vægtet to-delt pardannelse* angives kun, hvilke par (A_i , B_j), der er mulige, sådan at alle sådanne par har samme vægt).

På figuren

	B_1	B_2	B_3	B_4	B_5
A_1	3	4	5	2	1
A_2	4	5	5	3	1
A_3	4	4	5	1	2
A_4	4	2	4	2	2
A_5	3	2	5	3	3

angiver pladserne med cirkler en job-tilordning, hvor ansøger A_1 får job B_5 , A_2 får B_5 , o.s.v. (eller de angiver en pardannelse med par (A_1 , B_5) , (A_2 , B_3) , o.s.v.). Den samlede vægt af pladserne med cirkler er 12.

Er 12 det minimalt opnåelige på 5 skæve pladser ? Svare et NEJ. En letforklarlig begrundelse herfor er, at følgende 5 skæve pladser kun har sum 11 :

	B_1	B_2	B_3	B_4	B_5
A_1	3	4	5	2	1
A_2	4	5	5	3	1
A_3	4	4	5	1	2
A_4	4	2	4	2	2
A_5	3	2	5	3	3

Er 11 det minimalt opnåelige på 5 skæve pladser ? Svaret er JA. En letforklarlig begrundelse herfor er, at de 5 givne pladser med sum 11 består af et minimalt element fra hver søjle, så en mindre sum kan aldrig opnås på 5 pladser med én plads fra hver søjle.

I eksemplet ovenfor så vi, at både da svaret var NEJ og da det var JA, eksisterede der en letforklarlig begrundelse for det givne svar. En *god sætning* for en problem-type er en karakterisation, som i alle mulige tilfælde giver en letforklarlig begrundelse for det korrekte svar.

I §II.3 skal vi finde en god sætning for job-tilordnings-problemet. Det betyder, at der altid for et sådant problem eksisterer en letforklarlig begrundelse for det korrekte svar. For elever, som skal finde n skæve pladser med minimal sum i en givet matrix, er det ikke nødvendigvis så let at finde det korrekte svar (der er n! muligheder). Men er svaret først fundet, så er der også en letforklarlig begrundelse for det. At der er en god sætning betyder altså, at det er let for læreren at kontrollere, at et givet svar er korrekt.

At der er en god sætning betyder således ikke umiddelbart, at der er en god algoritme til at finde det korrekte svar. Imidlertid vil beviset for den gode sætning for job-tilordning samtidig give os en god algoritme. Altså er også job-tilordning et eksempel på, at gode algoritmer og gode sætninger tilsyneladende følges ad, jvf. side 1.4 nederst.

Før vi betragter job-tilordning, skal vi i §II.2 se på ikke-vægtet to-delt pardannelse. I §II.4 og II.5 ser vi på mere generelle problem-typer, specielt *lineær programmering* (LP), hvor den såkaldte dualitetssætning er en god sætning, og *transport-problemet*.

De typer af pardannelse, som optrådte i ALGORITME POSTBUD og i ALGORITME HANDELSREJSENDE, var vægtet, men <u>ikke</u> todelt, idet der ikke skulle dannes par af to forskellige typer af elementer. Også denne type af pardannelse er løst, både i form af gode algoritmer og gode sætninger, men vi skal ikke behandle dem her, da løsningerne (udviklet af den canadiske matematiker Jack Edmonds i 1960'erne) er ganske komplicerede.

II.2 IKKE VÆGTET, TO-DELT PARDANNELSE

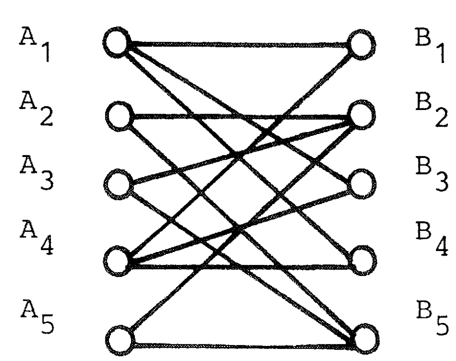
Eksempel 2.2 I et selskab med 5 drenge $A_1, A_2, \ldots, A_5$ ønsker vi at finde en passende pige til så mange af drengene som muligt. A_1 ønsker kun at danne par med B_1, B_3 eller B_4. Drengen A_2 er kræsen: han ønsker kun at danne par med B_2 eller B_5. Og A_3 ligeledes kun med B_2 eller B_5. Og A_4 kun med B_1, B_3 eller B_4. Og endelig A_5 kun med B_2 eller B_5 (pigernes mening interesserer vi os ikke for!).

Situationen kan beskrives v.h.a. matricen

	B_1	B_2	B_3	B_4	B_5
A_1	0		0	0	
A_2		0			0
A_3		0			0
A_4	0		0	0	
A_5		0			0

hvor vi ønsker at finde så mange med 0 mærkede skæve pladser som muligt.

Alternativt kan situationen beskrives v.h.a. den *to-delte* graf



hvor vi ønsker at finde så mange *uafhængige* kanter (dvs. kanter uden fælles endepunkter) som muligt.

At det i den givne situation <u>ikke</u> er muligt at danne 5 par kan ses af, at A_1, A_4, B_2 og B_5 *dækker* alle de mulige par, d.v.s. alle de mulige par indeholder mindst én af disse 4, så der kan højst dannes 4 par. Alternativt kunne vi sige, at de 3 drenge A_2, A_3 og A_5 tilsammen kun er interesseret i de to piger B_2 og B_5, og at det derfor <u>ikke</u> er muligt at opnå en pardannelse, som imødekommer alle 5 drenge.

De letforklarlige grunde, som blev givet i Eksempel 2.2 gælder generelt for to-delt ikke vægtet pardannelse. Dette blev vist af henholdsvis ungaren D. König i 1931 og af englænderen P. Hall i 1935. Vi skal nedenfor gengive König's bevis, som er konstruktivt.

Idéen i König's bevis er for en givet to-delt graf at starte med en tilfældig pardannelse (kanterne i denne tegner vi tykke). Såfremt der i grafen eksisterer en *alternerende* vej, dvs. en vej af typen

mellem to uparrede punkter x og y, så kan vi ved på denne vej at fjerne alle de tykke kanter fra pardannelsen og tilføje alle de tynde kanter til pardannelsen, opnå en ny pardannelse med ét par mere end før.

Da den givne graf er to-delt, vil punkterne på en vej ligge skiftevis i den ene og den anden side, specielt vil endepunkterne x og y af en alternerende vej ligge i hver sin side.

Hjertet i König's bevis er en systematisk metode til at opdage sådanne alternerende veje. Efter formuleringen af algoritmen skal vi vise, at den virkelig giver det beskrevne output, og at den implicerer de gode sætninger af König og Hall.

ALGORITME PARDANNELSE

INPUT: To-delt graf G med punkter
$\{A_1, A_2, \ldots, A_n\}$ og $\{B_1, B_2, \ldots, B_m\}$
i de to sider.

OUTPUT: En størst mulig pardannelse P i G,
dvs. flest mulige uafhængige kanter i G.

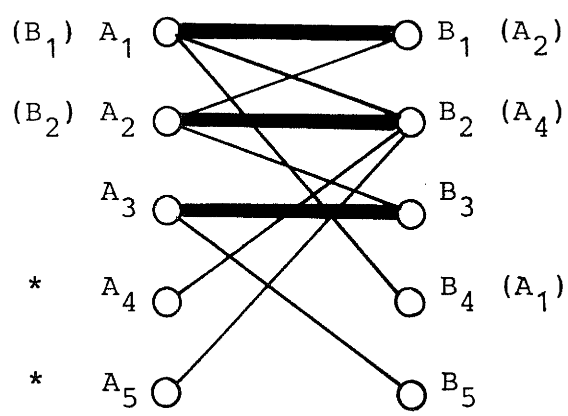
BEGIN
P : = en vilkårlig pardannelse i G, f.eks. $\emptyset$;
fortsæt : = sand ;
WHILE fortsæt DO
BEGIN
Mærk alle punkter i $\{A_1, A_2, \ldots, A_n\}$ ikke incident med kant fra P med * ; mærk punkterne i $\{B_1, B_2, \ldots, B_m\}$ og $\{A_1, A_2, \ldots, A_n\}$ skiftevis (indtil ikke flere punkter kan mærkes) efter reglerne

a) for alle ny-mærkede A_i mærkes med (A_i) alle ikke-mærkede B_j forbundet til A_i med kant i G,

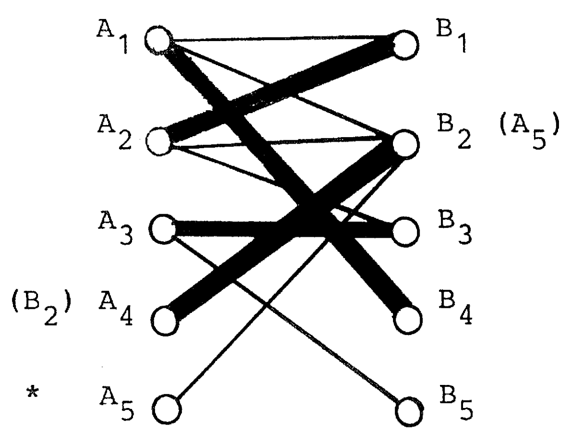
b) for alle ny-mærkede B_j mærkes med (B_j) alle ikke-mærkede A_i forbundet til B_j med kant i P (* læg mærke til, at der stod P *);

IF der er et mærket punkt B_j, som ikke er incident med en kant i P
THEN BEGIN
Følg mærkerne fra B_j tilbage til et punkt mærket * og opnå derved en alternerende vej V ;
P : = pardannelsen opnået fra P og V ved at skifte på V ;
Fjern alle mærker
END
ELSE fortsæt : = falsk
END
END.

Eksempel 2.3 Som et eksempel på mærkningsprocessen betragter vi følgende graf, hvor de tykke kanter er pardannelsen P , som vi starter med. Mærkningen i WHILE-løkken giver f.eks. (mærkningen afhænger af rækkefølgen ny-mærkede punkter betragtes i):



Vejen med punkter $B_4, A_1, B_1, A_2, B_2, A_4$ i denne rækkefølge udgør en alternerende vej V. Ved at skifte på denne vej opnås en ny pardannelse. Ved ændring af P og et nyt gennemløb af WHILE-løkken fås mærkningen:

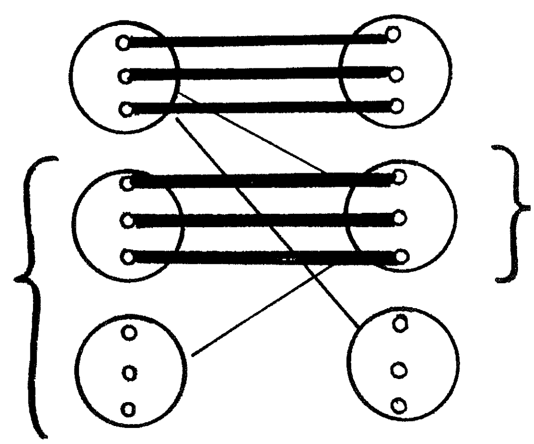


og i IF-THEN-ELSE-ordren havner vi i ELSE, og algoritmen standser.

Opgave 2.1 Gennemfør algoritmen på forskellige to-delte grafer, du selv tegner.

ALGORITME PARDANNELSE er en god algoritme. Et gennemløb af WHILE-løkken ender enten med, at pardannelsen P erstattes af en pardannelse med ét par mere, eller med, at algoritmen standser. WHILE-løkken kan derfor højst gennemløbes n gange. I WHILE-løkken benyttes hver kant højst et konstant antal gange uafhængig af grafens størrelse. Alt i alt er algoritmen af kompleksitet højst $O(n^2m)$, dvs. $O(|V(G)|^3)$.

Tilbage står nu at vise, at ALGORITME PARDANNELSE virkelig giver det ønskede output. Algoritmen standser med en pardannelse P og en mærkning:



De mærkede B_j'er er alle incidente med kanter fra P. Endvidere er de mærkede A_i'er præcis de mærkede B_j'ers partnere i P forenet med alle A_i'er ikke incidente med kanter i P, se figuren ovenfor. Da mærkningen er gået i stå, er der ingen kanter mellem et mærket A_i og et ikke mærket B_j. Altså *dækker* de ikke mærkede A_i'er og de mærkede B_j'er alle kanter i grafen, dvs. enhver kant i grafen har et af disse punkter som et endepunkt. Heraf følger, at en vilkårlig pardannelse har højst (antal ikke mærkede A_i'er + antal mærkede B_j'er) par. Men P har netop dette antal par, så P er størst mulig. Altså giver algoritmen det ønskede output.

For en to-delt graf G, en vilkårlig pardannelse P i G og en vilkårlig punktmængde D i G, som dækker alle kanter i G, gælder at $|P| \leq |D|$. ALGORITME PARDANNELSE finder et P og et D (nemlig D lig med de ikke mærkede A_i'er forenet med de mærkede B_j'er), således at $|P| = |D|$. Det fundne P er derfor en størst mulig pardannelse og det fundne D en mindst mulig dækning. Altså

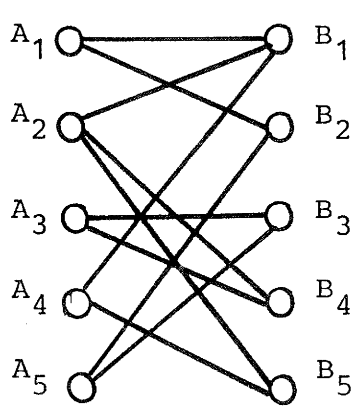
Sætning 2.1 (König 1931) Lad G være en to-delt graf. Så er

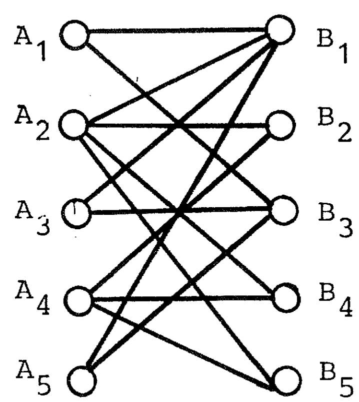
$$\max |P| = \min |D|$$

hvor P er en pardannelse i G, dvs. en mængde af kanter uden fælles punkter, og hvor D er en dækning i G, dvs. en mængde af punkter, så hver kant i G har mindst ét endepunkt i D.

König's sætning er et typisk eksempel på en god sætning. Antag, at vi har en to-delt graf G og spørger, om G har en pardannelse med p par. Hvis svaret er JA, så er der en letforklarlig begrundelse, nemlig fremvisningen af en pardannelse med p par. Hvis svaret er NEJ, så er der også en letforklarlig begrundelse, nemlig fremvisningen af en dækkende mængde D af punkter med $|D| \leq p-1$. König's sætning viser altså, at der er en letforklarlig begrundelse i alle tilfælde.

Opgave 2.2 Har følgende to grafer pardannelser med 5 par ? (Begrund svarene!).





Opgave 2.3 Vis vha. eksempler, at Sætning 2.1 ikke er sand for alle grafer.

Spørger vi, om der eksisterer en pardannelse i en to-delt graf G, så alle punkter i den ene side er med i et par (dvs. i terminologien fra Eksempel 2.2 : en pardannelse, som imødekommer alle drenge), så kan vi give den gode sætning følgende formulering:

<u>Sætning 2.2 (Hall 1935)</u> Lad G være en to-delt graf med punkter $\{A_1, A_2, \ldots, A_n\}$ og $\{B_1, B_2, \ldots, B_m\}$ i de to sider. <u>Enten</u> har G en pardannelse med n par, <u>eller</u> der eksisterer en delmængde af $\{A_1, A_2, \ldots, A_n\}$ som tilsammen er forbundet med kanter til en skarpt mindre delmængde af $\{B_1, B_2, \ldots, B_m\}$.

I terminologien fra Eksempel 2.2 lyder Sætning 2.2 således: <u>Enten</u> er der en pardannelse, som imødekommer alle drenge, <u>eller</u> der eksisterer en mængde af drenge, som tilsammen er interesserede i en mindre mængde af piger.

<u>Bevis for Sætning 2.2</u> Vi benytter igen ALOGRITME PARDANNELSE, hvor vi antager, at algoritmen standser med en pardannelse P med $|P| < n$.

De mærkede A_i'er er tilsammen forbundet med kanter til de mærkede B_j'er, som udgør en skarpt mindre mængde (se den seneste figur ovenfor). ∎

I Eksempel 2.2 så vi, at ikke vægtet, to-delt pardannelse kan formuleres både vha. to-delte grafer og vha. matricer. Sætningerne 2.1 og 2.2 formuleret for n×n-matricer ser således ud:

<u>Sætning 2.3</u> Lad M være en matrice med n rækker og n søjler. Lad nogle af de n^2 pladser være udvalgte (skriv f.eks. 0 på de udvalgte pladser).

Så er

$$\max |P| = \min |D|$$

hvor P er en *skæv* mængde af udvalgte pladser (dvs. P har højst ét element fra hver række og fra hver søjle), og hvor D er en *dækkende* mængde af rækker og søjler (d.v.s. enhver udvalgt plads står i enten en række eller en søjle fra D).

<u>Sætning 2.4</u> Lad M være en matrice med n rækker og n søjler. Lad nogle af de n^2 pladser være udvalgte (skriv f.eks. 0 på de udvalgte pladser).

Så eksisterer der <u>enten</u> n skæve udvalgte pladser <u>eller</u> en delmængde af rækkerne, hvor de udvalgte pladser i disse rækker tilsammen står i en mindre mængde af søjler.

Den følgende sekvens af opgaver munder ud i en anvendelse af pardannelse indenfor skemalægning.

<u>Opgave 2.4</u> Lad G være en to-delt graf med henholdsvis n og m punkter i de to sider. Antag, at alle punkter i G har samme valens r.

a) Vis, at $n = m$.
b) Vis, vha. Sætning 2.2, at der er en pardannelse i G med n par.
c) Vis, at G's kanter kan farves med r farver (hver kant får én farve), således at to kanter med et fælles endepunkt altid får forskellige farver (vink: induktion efter r).

<u>Opgave 2.5</u> Lad G være en to-delt graf, hvor den største valens er r.

a) Vis, at G kan udvides vha. nye punkter og kanter til en to-delt graf H, hvor alle punkter har valens lig med r.

b) Vis, v.ha. Opgave 2.4.c) og a) ovenfor, at G's kanter kan farves med r farver (som i Opgave 2.4.c)).

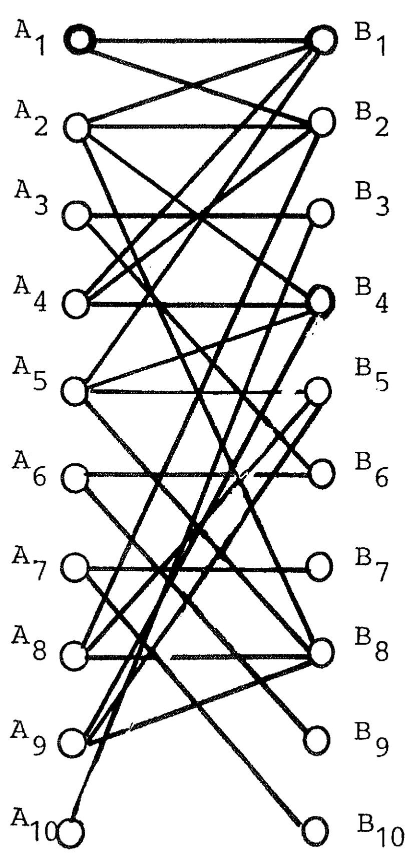
c) Vis, vha. et eksempel, at det er en nødvendig forudsætning for b), at G er to-delt.

<u>Opgave 2.6</u> Du skal lægge skema for en skole. Der er ialt 40 klasser, som hver skal have $\leq$ 30 t. pr. uge (og der er 7 klasser, som hver skal have præcis 30 t). Der er ialt 60 lærere på skolen, og det er allerede fastlagt, hvilke lærere, der skal have hvilke klasser til hvad. På grund af timeloft har ingen lærer over 27 t

pr. uge. Skolen har tilstrækkeligt med lokaler, og rektor beder dig lægge et skema, som afvikler samtlige timer på kortest mulig tid (vi antager, at hver skoletime har varighed 1 t).
Kan du på forhånd oplyse, hvor lang tid denne kortest mulige tid er ?
Vink: Formulér problemet som et kantfarvningsproblem og benyt Opgave 2.5.b).

Opgave 2.7 Find en størst mulig pardannelse P og en mindst mulig dækning D for grafen nedenfor. Find en størst mulig skæv mængde P og en mindst mulig dækkende mængde af rækker og søjler i matricen nedenfor.

Såfremt $|P| < 10$ angiv da for grafen en delmængde af $\{A_1, A_2, \dots, A_{10}\}$, som tilsammen er forbundet med kanter til en mindre delmængde af $\{B_1, B_2, \dots, B_{10}\}$, og for matricen en delmængde af rækkerne, hvori de udvalgte pladser tilsammen står i en mindre delmængde af søjlerne.



	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9	B_{10}
A_1	0	0								
A_2	0	0		0				0		
A_3			0			0				
A_4	0	0		0						
A_5	0			0	0			0		
A_6						0			0	
A_7							0			0
A_8		0			0			0		
A_9				0	0			0		
A_{10}			0							

II.3 JOB-TILORDNING

Et eksempel på *job-tilordnings-problemet* blev givet i Eksempel 2.1. I et problem af denne type er der givet en $n \times n$-*matrix*, dvs. en kvadratisk tabel af tal med n rækker og n søjler. Problemet er ud af de ialt n^2 pladser at finde n *skæve* pladser, dvs. n pladser, hvor ikke to er i samme række eller søjle, således at summen af tallene på de n pladser er mindst mulig. Grunden til navnet job-tilordning er, at man kan forestille sig, at rækkerne repræsenterer n ansøgere og søjlerne n jobs, og at tallet c_{ij} i række i og række j repræsenterer ansøger i's uegnethed til job j. En mængde af n skæve pladser svarer så til en tilordning af de n jobs til de n ansøgere, og en tilordning med mindst mulig sum svarer til en tilordning med mindst mulig samlet uegnethed.

Mere matematisk formuleret er problemet at finde en permutation P af mængden $\{1,2,\dots,n\}$ så

$$\sum_{i=1}^{n} c_{iP(i)}$$

er mindst mulig. Da der er ialt $n!$ mulige permutationer, er en løsning, hvor alle muligheder undersøges, alt for tidskrævende, selv for små værdier af n.

Opgave 2.8 Vis, at der ved undersøgelse (uden "smarthed") af alle muligheder, én efter én, i et job-tilordnings-problem af størrelse $n \times n$, skal foretages ialt $(n-1)\cdot(n!)$ sammenlægninger. Hvor lang tid vil dette kræve for $n=10$ med én sammenlægning pr. sekund ?

Der er altså behov for en mere effektiv metode end at prøve alle muligheder. Før vi viser, at en sådan eksisterer, er det måske på sin plads at antyde, at problemet ikke er helt let! Dette sker i de følgende to opgaver.

<u>Opgave 2.9</u> Overvej forskellige simple algoritmer, f.eks.

1) Giv A_1 det job, han er bedst til. Giv så A_2 det af de resterende jobs, han er bedst til, osv.

2) Lad c_{ij} være det mindste tal. Giv A_i jobbet B_j. Fjern række i og søjle j fra matricen. Fortsæt.

Vis ved eksempler, at sådanne grådige algoritmer ikke dur.

<u>Opgave 2.10</u> Betragt problemet med matrix

	B_1	B_2	B_3
A_1	1	3	4
A_2	3	7	8
A_3	2	5	7

De indcirklede pladser er tre skæve pladser med mindst mulig sum.

Antag, at en fjerde ansøger

	B_1	B_2	B_3
A_4	9	8	6

melder sig. A_4 kan klare hvert af jobbene dårligere end nogen af de tre, vi har tilordnet til disse job. Skal vi sige til A_4, at jobbene desværre er optaget, eller skal vi hellere antage A_4 og lade f.eks. A_2 gå ?

Nøglen til en løsning af job-tilordnings-problem-typen ligger i følgende observation: <u>Ved at trække det samme tal α fra alle pladser i en række (eller en søjle) fås en matrix med præcis de samme n skæve pladser med mindst mulig sum som den oprindelige matrix (vi vil sige, at de to matricer er *ækvivalente*)</u>. Dette følger af, at enhver mængde af n skæve pladser i den nye matrix har en sum, som er α mindre end summen af de samme n skæve pladser i den oprindelige matrix.

En mulig metode til løsning af problemer af denne type er derfor, at simplificere matricen ved at trække tal fra i rækker og søjler. Hvis vi gør det sådan, at alle tal i den resulterende ækvivalente matrix er $\geqq 0$, og sådan, at der er n skæve pladser med lutter 0'er, så har disse n skæve pladser klart mindst mulig sum i den nye matrix og dermed også i den oprindelige.

Opgave 2.11 Prøv at løse job-tilordnings-problemet

	B_1	B_2	B_3	B_4
A_1	9	8	8	7
A_2	8	7	7	5
A_3	7	7	6	5
A_4	7	6	5	4

med metoden ovenfor. Vis, at der er præcis én tilordning med mindst mulig sum.

Som vi skal se i det følgende, virker strategien ovenfor altid. Problemet er blot at finde en systematisk metode til at finde de tal $\alpha_1, \alpha_2, \ldots, \alpha_n$, som skal trækkes fra i de n rækker, og de tal $\beta_1, \beta_2, \ldots, \beta_n$, som skal trækkes fra i de n søjler. Disse tal skal opfylde

(*) $\quad c_{ij} - \alpha_i - \beta_j \geqq 0$ for alle i og j.

(**) Der eksisterer n skæve pladser blandt pladserne (i,j) med $c_{ij} - \alpha_i - \beta_j = 0$.

I stedet for at ændre matricen ved at trække α_i'er og β_j'er fra, er det bedre (for at undgå regnefejl), at bevare den oprindelige matrix, skrive tallene $\alpha_1, \alpha_2, \ldots, \alpha_n$ ud for de n rækker, og skrive tallene $\beta_1, \beta_2, \ldots, \beta_n$ over de n søjler, hvor vi så hele tiden kan holde øje med, at

(*) $\quad \alpha_i + \beta_j \leqq c_{ij}$ for alle i og j

er opfyldt. De pladser (i,j), hvor $\alpha_i + \beta_j = c_{ij}$, markeres, f.eks. med ◯. Det er blandt disse pladser, vi skal finde n skæve, jvf. (**).

For rækkevægte $\alpha_1, \alpha_2, \ldots, \alpha_n$ og søjlevægte $\beta_1, \beta_2, \ldots, \beta_n$, som opfylder (*), gælder for ethvert P af n skæve pladser:

$$(***) \qquad \sum_{i=1}^{n} \alpha_i + \sum_{j=1}^{n} \beta_i \leqq \sum_{(i,j)\in P} c_{ij}$$

(***) vises ved at summere (*) for alle $(i,j) \in P$. Summen af vægtene er altså en nedre grænse for, hvor langt ned vi kan nå med summen af n skæve pladser.

Lykkes (**), så gælder der lighedstegn i (***) for de pågældende α_i'er, β_j'er og n skæve pladser P. Analysen ovenfor viser, at et sådant P består af n skæve pladser med mindst mulig sum. Dette følger også af lighedstegnet i (***), idet vi så med $\sum_{(i,j)\in P} c_{ij}$ er helt nede på den nedre grænse $\sum_i \alpha_i + \sum_j \beta_j$. (***) viser samtidig, at de pågældende α_i'er og β_j'er har størst mulig samlet sum.

Alt afhænger altså af om (**) altid kan lykkes. Vi skal se, at dette er tilfældet.

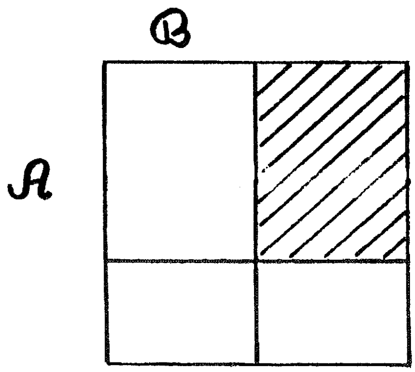
Eksempel 2.4 For matricen i Opgave 2.11 kan følgende rækkevægte og søjlevægte, som opfylder (*), angives. Pladserne med $= \alpha_i + \beta_j = c_{ij}$ er markeret med ○.

$$\begin{array}{c|cccc|}
 & 2 & 1 & 1 & 0 \\
\hline
7 & 9 & 8 & 8 & 7 \\
5 & 8 & 7 & 7 & 5 \\
5 & 7 & 7 & 6 & 5 \\
4 & 7 & 6 & 5 & 4 \\
\hline
\end{array}$$

Pladserne med ○✓ er den eneste måde at udvælge 4 skæve af de markerede pladser. Disse giver derfor den entydigt bestemte løsning på job-tilordnings-problemet.

Læg mærke til, at $\Sigma\alpha_i + \Sigma\beta_j = 25$, som derfor er en nedre grænse for summen af fire skæve pladser iflg.(***), og at summen af de fire angivne skæve pladser netop er lig med 25. De viste vægte er derfor en letforklarlig begrundelse for, at tilordningen $\{(A_1, B_2), (A_2, B_4), (A_3, B_1), (A_4, B_3)\}$ har mindst mulig sum.

Lad os nu forestille os, at vi er nået frem til rækkevægte $\alpha_1, \alpha_2, \ldots, \alpha_n$ og søjlevægte $\beta_1, \beta_2, \ldots, \beta_n$, som opfylder (*), men således at der desværre ikke eksisterer n skæve pladser blandt de med O markerede pladser, dvs. pladserne med $\alpha_i + \beta_j = c_{ij}$. Så sikrer Sætning 2.4 os, at der er en delmængde $\mathcal{A}$ af rækkerne, hvor de markerede pladser i disse rækker står i en mindre mængde $\mathcal{B}$ af søjler:



I det skraverede område er der altså ingen med O markerede pladser, dvs. $\alpha_i + \beta_j < c_{ij}$ for alle disse pladser. Lad $\delta = \min(c_{ij} - \alpha_i - \beta_j)$, hvor minimum tages over alle pladser i det skraverede område. For rækkerne i $\mathcal{A}$ kan nu δ lægges til de tilsvarende α_i'er, og for rækkerne i $\mathcal{B}$ kan δ trækkes fra de tilsvarende β_j'er. Herved fås et sæt nye vægte, som stadig opfylder (*), men hvor $\Sigma\alpha_i + \Sigma\beta_j$ er større end før (da δ lægges til flere gange end det trækkes fra). Der opstår mindst én ny med O markeret plads i det skraverede område, og vi kan igen lede efter n skæve pladser blandt de nu med O markerede pladser.

Ovenstående idéer kan sammenfattes i følgende algoritme:

ALGORITME JOB-TILORDNING

INPUT : $n \times n$-matrix med reelle tal på de n^2 pladser (c_{ij} i række i og søjle j).

OUTPUT: Mængde P af n skæve pladser med $\sum_{(ij) \in P} c_{ij}$ mindst mulig blandt alle mulige P.

BEGIN

Find rækkevægte $\alpha_1, \alpha_2, \ldots, \alpha_n$ og søjlevægte $\beta_1, \beta_2, \ldots, \beta_n$ således at $\alpha_i + \beta_j \leqq c_{ij}$ for alle i og j, f.eks. α_i = (mindste tal i række i) og $\beta_j = 0$;

fortsæt : = sand;

WHILE fortsæt DO

BEGIN

Markér med ○ alle pladser (i,j), hvor $\alpha_i + \beta_j = c_{ij}$;

Find vha. ALGORITME PARDANNELSE en størst mulig mængde P af skæve med ○ markerede pladser ;

IF $|P| < n$

THEN BEGIN

Find delmængde $\mathcal{A}$ af rækkerne, hvor de med ○ markerede pladser står i en mindre mængde $\mathcal{B}$ af søjler (* jvf. Sætningerne 2.2 og 2.4. $\mathcal{A}$ og $\mathcal{B}$ er iflg. beviset for Sætning 2.2 de i ALGORITME PARDANNELSE mærkede A_i'er og B_j'er *);

$$\delta := \min_{\substack{i \in \mathcal{A} \\ j \notin \mathcal{B}}} (c_{ij} - \alpha_i - \beta_j);$$

$\alpha_i := \alpha_i + \delta$ for alle $i \in \mathcal{A}$;

$\beta_j := \beta_j - \delta$ for alle $j \in \mathcal{B}$

END

ELSE fortsæt := falsk

END

END.

Hvis alle c_{ij}'er er heltallige, så slutter algoritmen i et endeligt antal skridt. Det følger af, at vi i dette tilfælde også kan forudsætte, at α_i'erne og β_j'erne er heltallige, at $\Sigma\alpha_i + \Sigma\beta_j$ bliver mindst 1 større ved hvert gennemløb af THEN ordren, samt at der er en øvre grænse på $\Sigma\alpha_i + \Sigma\beta_j$, f.eks. Σc_{ii}.

I det følgende antydes argumentet for at ALGORITME JOB-TILORDNING altid standser, og at der faktisk er tale om en god algoritme. Afsnittet er kortfattet og derfor måske svært. Det kan springes over.

↓ Ved efter ét gennemløb af THEN-ordren at lade mærkningen i ALGORITME PARDANNELSE fortsætte, hvor den var nået til forrige gang, vil et nyt B_j blive mærket. Heraf følger, at THEN-ordren skal udføres højst n gange inden $|P|$ bliver større. Antal skridt mellem to på hinanden følgende forøgelser af $|P|$ bliver derfor højst $O(n \cdot n^2)$, idet det dominerende arbejde er højst n gange at beregne et nyt δ som minimum af højst n^2 tal. Da $|P|$ bliver større højst n gange, er ALGORITME PARDANNELSE derfor af kompleksitet højst $O(n^4)$, dvs. der er tale om en god algoritme.

Nogle af tallene, som skal beregnes for at finde δ, går igen ved næste beregning af δ. Disse beregninger kan derfor effektiviseres, og de kan udføres, så ALGORITME JOB-TILORDNING får
↑ kompleksitet højst $O(n^3)$.

Vi har nu overbevist os om, at ALGORITME JOB-TILORDNING virkelig er en algoritme, som standser efter ikke alt for mange skridt.

At ALGORITME JOB-TILORDNING også giver det beskrevne output, kan indses således: For rækkevægte $\alpha_1, \alpha_2, \ldots, \alpha_n$ og søjlevægte $\beta_1, \beta_2, \ldots, \beta_n$, som opfylder $\alpha_i + \beta_j \leqq c_{ij}$ for alle i og j, og for n skæve pladser P, gælder som tidligere set altid

$$(***) \qquad \Sigma\alpha_i + \Sigma\beta_j \leqq \sum_{(i,j)\in P} c_{ij}$$

For slutvægtene $\alpha_1, \alpha_2, \ldots, \alpha_n, \beta_1, \beta_2, \ldots, \beta_n$ og det afsluttende P i ALGORITME JOB-TILORDNING gælder = i (***). Der er derfor tale om et P bestående af n skæve pladser med $\sum_{(ij)\in P} c_{ij}$

mindst mulig, som ønsket. ∎

Ovenstående bevis for at ALGORITME JOB-TILORDNING giver det korrekte output beviser også følgende gode sætning:

<u>Sætning 2.5</u> For en $n \times n$-matrix M med tal c_{ij} i række i og søjle j, gælder

$$\max \left(\sum_{i=1}^{n} \alpha_i + \sum_{j=1}^{n} \beta_j \right) = \min \sum_{(i,j) \in P} c_{ij}$$

hvor maximum tages over alle $\alpha_1, \alpha_2, \ldots, \alpha_n, \beta_1, \beta_2, \ldots, \beta_n$, som opfylder $\alpha_i + \beta_j \leqq c_{ij}$ for alle i og j, og hvor minimum tages over alle mængder P af n skæve pladser.

Sætning 2.5 er god. Antag, at vi spørger, om M har n skæve pladser med sum $\leqq s$. Hvis svaret er JA, så er der en letforklarlig begrundelse, nemlig fremvisningen af n skæve pladser med sum $\leqq s$. Hvis svaret er NEJ, så er der også en letforklarlig begrundelse, nemlig en fremvisning af rækkevægte $\alpha_1, \alpha_2, \ldots, \alpha_n$ og søjlevægte $\beta_1, \beta_2, \ldots, \beta_n$ så $\alpha_i + \beta_j \leqq c_{ij}$ og $\Sigma\alpha_i + \Sigma\beta_j > s$.

ALGORITME JOB-TILORDNING blev udviklet i begyndelsen af 1950'erne af den amerikanske økonomom H.W. Kuhn. Da ALGORITME PARDANNELSE spiller en væsentlig rolle, kaldte Kuhn fremgangsmåden *den ungarske metode* til ære for D. König.

Lad os bemærke, at selv om vi i alle de tidligere eksempler har ladet c_{ij}'erne være hele, positive tal, så har dette <u>ikke</u> været en nødvendighed. Den ungarske metode virker for vilkårlige reelle tal c_{ij}, også negative. Denne bemærkning kan udnyttes, hvis vi stilles overfor opgaven i en $n \times n$-matrix at finde n skæve pladser med <u>størst</u> mulig sum. Så ændres blot fortegn på alle tal i matricen, og n skæve pladser med mindst mulig sum i den nye matrix har størst mulig sum i den oprindelige.

Opgave 2.12 Find for 5×5-matricen

12	9	10	8	11
8	6	6	5	9
13	10	10	11	11
6	2	4	3	5
11	7	10	9	11

5 skæve pladser med mindst mulig sum og 5 skæve pladser med størst mulig sum.

Giv letforklarlige grunde til, at de opnåede løsninger er korrekte.

Job-tilordnings-problemet og Sætning 2.5 kan gives en økonomisk fortolkning. Tallene c_{ij} kan betegne den løn, som vi skal betale ansøgeren A_i for at udføre job B_j. Job-tilordnings-problemet er så for os at finde en tilordning med mindst mulig samlet lønudgift. Lad os nu forestille os, at ansøgerne slutter sig sammen og tilbyder os, at de selv vil finde ud af, hvem der skal gøre hvad. Vi skal blot for alle i udbetale A_i et beløb på α_i og for alle j betale β_j, når job B_j er udført. For at overbevise os om det fornuftige i denne sidste løsning, lader ansøgerne tallene α_i og β_j opfylde $\alpha_i + \beta_i \leqq c_{ij}$ for alle i og j, så det alt i alt ser ud til at blive billigere for os. På den anden side er ansøgerne naturligvis interesserede i at få så stort samlet udbytte som muligt, dvs. $\Sigma\alpha_i + \Sigma\beta_j$ skal være så stor som muligt. Sætning 2.5 siger, at de optimale løsninger på de to problemer giver samme udgift for os.

II.4. LINEÆR PROGRAMMERING OG DUALITET.

Sætning 2.5 er et godt eksempel på en maximum-minimum-sætning. Den kendteste maximum-minimum-sætning i matematik er den såkaldte dualitets-sætning i lineær programmering (LP). Vi skal i dette afsnit forklare denne sætning (men ikke bevise den i alle detaljer), og vi skal se, at Sætning 2.5 kan betragtes som et special-tilfælde.

Eksempel 2.5

Vi skal sammensætte en foderblanding af tre ingredienser U , V og W. Prisen pr. kg er henholdsvis 30 kr , 35 kr og 20 kr. Den færdige blanding skal indeholde mindst 50 enheder af stoffet S_1 , hvor ét kg af U , V og W indeholder henholdsvis 2, 7 og 5 enheder. Den færdige blanding skal endvidere indeholde mindst 40 enheder af stoffet S_2 , hvor ét kg af U , V og W indeholder henholdsvis 5, 5 og 2 enheder. Vi ønsker en billigst mulig blanding. 25 kg af U ville være nok til at sikre de ønskede mængder af S_1 og S_2, dvs. vi kan klare det for 750 kr. 1 kg af U og 7 kg af V ville også være nok, og så kommer prisen ned på kun 305 kr. Der er mange muligheder! At prøve sig frem løser nok ikke problemet. Vi må være systematiske! Problemet på matematisk form ser således ud, idet u , v og w betegner det antal kg af U , V og W , som vi skal blande:

Primære problem

Find min $30u + 35v + 20w$

ufa.
$$2u + 7v + 5w \geqq 50$$
$$5u + 5v + 2w \geqq 40$$
$$u \geqq 0 \ , \ v \geqq 0 \ , \ w \geqq 0.$$

Dette er et typiske LP-*problem*, hvor vi skal finde minimum (eller maximum) af en *lineær* funktion af flere variable, dvs. en funktion af typen $b_1y_1 + b_2y_2 + \dots + b_ny_n$ (hvor b_i'erne er konstanter og y_i'erne de variable), under forudsætning af (forkortet ufa.) nogle lineære uligheder (og/eller ligheder).

I stedet for at kaste os ud i en løsning af problemet, vil vi i første omgang forsøge at finde nedre grænser for, hvor lille $30u + 35v + 20w$ kan blive.

Hvis vi f.eks. ganger den første ulighed med 4 fås

$$4 \cdot 50 \leqq 4 \cdot (2u + 7v + 5w) \leqq 30u + 35v + 20w$$

hvor det sidste ulighedstegn benytter, at $u \geqq 0$, $v \geqq 0$ og $w \geqq 0$. Tallet 200 er derfor en nedre grænse for $30u + 35v + 20w$.

Hvis vi ganger den anden ulighed med 6 fås tilsvarende

$$6 \cdot 40 \leqq 6 \cdot (5u + 5v + 2w) \leqq 30u + 35v + 20w$$

dvs. vi får den bedre nedre grænse 240.

Ved at kombinere de to uligheder kan vi yderligere forbedre den nedre grænse. Ved at gange den første ulighed med 1 og den anden med 5 og lægge sammen fås:

$$50 \cdot 1 + 40 \cdot 5 \leqq (2u + 7v + 5w) \cdot 1 + (5u + 5v + 2w) \cdot 5 =$$

$$27u + 32v + 15w \leqq 30u + 35v + 20w ,$$

hvilket giver, at 250 er en nedre grænse.

Vi har altså nu en nedre grænse på 250, og tidligere havde vi en øvre grænse på 305. Kan vi formindske gabet yderligere med den teknik, vi benytter ? Svaret er JA. Vi kan faktisk med teknikken helt lukke gabet, ikke blot i det specielle LP-problem, vi betragter, men i ethvert LP-problem. Dette faktum kaldes *dualitetssætningen* for LP.

Hvad er den bedst mulige nedre grænse, vi kan opnå med teknikken ovenfor ? Lad os antage, at vi ganger den første ulighed med x og den anden med y . Så fås:

$$50x + 40y \leqq (2u + 7v + 5w)x + (5u + 5v + 2w)y =$$

$$(2x + 5y)u + (7x + 5y)v + (5x + 2y)w \leqq 30u + 35v + 20w ,$$

hvor det første ulighedstegn forudsætter, at $x \geqq 0$ og $y \geqq 0$, og hvor det sidste ulighedstegn forudsætter, at $2x + 5y \leqq 30$, $7x + 5y \leqq 35$ og $5x + 2y \leqq 20$.

Den bedst mulige nedre grænse fås altså som løsning på følgende problem:

<u>Duale problem</u>

$$\begin{array}{lll} \text{Find} & \max & 50x + 40y \\ & \text{ufa.} & 2x + 5y \leqq 30 \\ & & 7x + 5y \leqq 35 \\ & & 5x + 2y \leqq 20 \\ & & x \geqq 0 \ , \ y \geqq 0. \end{array}$$

Dette er et nyt LP-problem (!), som kaldes det *duale* problem af det oprindelige. Det oprindelige problem kaldes ofte i forhold til det duale problem for det *primære* problem. Løsninger (u,v,w) til det primære (eller (x,y) til det duale problem), som opfylder begrænsningerne, men som ikke nødvendigvis giver minimum (eller maximum), kaldes *brugbare*. Brugbare løsninger, som giver minimum (eller maximum) kaldes *optimale*.

Argumentationen ovenfor viser, at hvis (u,v,w) og (x,y) er brugbare løsninger til henholdsvis det primære og det duale problem, så er

$$50x + 40y \leqq 30u + 35v + 20w$$

Dette gælder specielt for optimale løsninger, altså har vi, at <u>maximum i det duale problem er</u> $\leqq$ <u>minimum i det primære problem</u>. Dette udsagn kaldes DEN SVAGE DUALITETSSÆTNING. Den tilsvarende STÆRKE sætning siger, at der faktisk gælder, at <u>maximum i det duale problem er lig med minimum i det primære problem</u>.

Det duale problem har en variabel for hver ulighed i det primære problem og en ulighed for hver variabel i det primære problem. Det duale af det duale problem er det primære problem.

For at indse den sidste påstand omformes det duale problem således, at det får samme form som det primære problem:

$$\begin{array}{lll} \text{Find} & \min & (-50x - 40y) \\ & \text{ufa.} & -2x - 5y \geqq -30 \\ & & -7x - 5y \geqq -35 \\ & & -5x - 2y \geqq -20 \\ & & x \geqq 0 \; , \; y \geqq 0. \end{array}$$

Herefter dannes det duale problem som før, og man får netop det primære problem. Prøv selv!

Såfremt vi accepterer den stærke dualitetssætning, har vi for optimale løsninger (u_o , v_o , w_o) og (x_o , y_o) til de to problemer, at

$$\begin{array}{ll} 50x_o + 40y_o & \leqq \\ (2u_o + 7v_o + 5w_o)\cdot x_o + (5u_o + 5v_o + 2w_o)\cdot y_o & = \\ (2x_o + 5y_o)\cdot u_o + (7x_o + 5y_o)v_o + (5x_o + 2y_o)w_o & \leqq \\ 30u_o + 35v_o + 20w_o & = \\ 50x_o + 40y_o \; , & \end{array}$$

altså gælder der lighedstegn hele vejen. Heraf sluttes:

$$\begin{array}{llll} \underline{\text{Enten}} & x_o = 0 & \underline{\text{eller}} & 2u_o + 7v_o + 5w_o = 50. \\ \text{"} & y_o = 0 & \text{"} & 5u_o + 5v_o + 2w_o = 40. \\ \text{"} & u_o = 0 & \text{"} & 2x_o + 5y_o = 30. \\ \text{"} & v_o = 0 & \text{"} & 7x_o + 5y_o = 35. \\ \text{"} & w_o = 0 & \text{"} & 5x_o + 2y_o = 20. \end{array}$$

Dette kan kort udtrykkes således: <u>For optimale løsninger gælder, at enten er en variabel 0 eller der er lighedstegn i den tilsvarende ulighed</u>. <u>Omvendt er brugbare løsninger, som opfylder dette, optimale</u>. Denne sætning kaldes SÆTNINGEN OM KOMPLEMENTÆR REST.

Såfremt f.eks. den første af ulighederne i det primære problem havde været en lighed, altså

$$2u + 7v + 5w = 50 \; ,$$

så ville ændringen i det tilsvarende duale problem blot være, at vi <u>ikke</u> behøvede at forudsætte, at den tilsvarende variabel

x er $\geqq 0$, dvs. x blev uden begrænsning. Eller var f.eks. u uden begrænsning i det primære problem, så ville ændringen i det duale problem være, at $2x + 5y \leqq 30$ blev erstattet af $2x + 5y = 30$. Dualitetssætningerne gælder stadigvæk i disse tilfælde.

Vejledt af Eksempel 2.5 kan følgende lille oversigt over tilsvarende elementer i det primære og det duale problem opstilles:

Primære	Duale
variabel $\geqq 0$	ulighed
ubegrænset variabel	lighed
ulighed	variabel $\geqq 0$
lighed	ubegrænset variabel

I Eksempel 2.5 gik vi stiltiende ud fra, at der er optimale løsninger til såvel det primære som det duale problem. Dette behøver ikke generelt at være tilfældet. Det kunne f.eks. være, at de brugbare løsninger i det primære problem kunne give værdier mindre end et vilkårligt givet tal, dvs. at problemet er *ubegrænset*. I så tilfælde er der ingen nedre grænse, og det betyder, at det duale problem slet ingen brugbare løsninger har, jvf. Eksempel 2.5. I formuleringen nedenfor af dualitetssætningen forudsætter vi, at vi ikke er i en sådan situation.

I dualitetssætningen er følgende størrelser givet på forhånd:

1) to hele positive tal m og n ;
2) en opdeling af $\{1,2,\dots,m\}$ i to mængder I og K , dvs. $I \cup K = \{1,2,\dots,m\}$ og $I \cap K = \emptyset$;
3) en opdeling af $\{1,2,\dots,n\}$ i to mængder R og F , dvs. $R \cup F = \{1,2,\dots,n\}$ og $R \cap F = \emptyset$;
4) reelle tal $b_1 , b_2 , \dots, b_m$;
5) reelle tal $c_1 , c_2 , \dots, c_n$;
6) reelle tal a_{ij} for $i=1,2,\dots,m$ og $j=1,2,\dots,n$.

<u>Sætning 2.6</u> <u>DUALITET FOR LP.</u> (J. von Neumann og G.B. Dantzig 1947; D. Gale, H.W. Kuhn, A.W. Tucker 1951).

For de duale problemer:

$$\begin{array}{ll} \text{Find min} & b_1y_1 + b_2y_2 + \dots + b_my_m \\ \text{ufa.} & a_{1j}y_1 + a_{2j}y_2 + \dots + a_{mj}y_m \geqq c_j \text{ for } j\in R \\ & a_{1j}y_1 + a_{2j}y_2 + \dots + a_{mj}y_m = c_j \text{ for } j\in F \\ & y_i \geqq 0 \text{ for } i\in I \text{ , } y_i \text{ ubegrænset for } i\in K \end{array}$$

og

$$\begin{array}{ll} \text{Find max} & c_1x_1 + c_2x_2 + \dots + c_nx_n \\ \text{ufa.} & a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \leqq b_i \text{ for } i\in I \\ & a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n = b_i \text{ for } i\in K \\ & x_j \geqq 0 \text{ for } j\in R \text{ , } x_j \text{ ubegrænset for } j\in F \end{array}$$

gælder

a) <u>Svag dualitet</u>
For brugbare løsninger $(y_1, y_2, \dots, y_m)$ og $(x_1, x_2, \dots, x_n)$ til de to problemer er

$$c_1x_1 + c_2x_2 + \dots + c_nx_n \leqq b_1y_1 + b_2y_2 + \dots + b_my_m$$

b) <u>Stærk dualitet</u>
Hvis der eksisterer brugbare løsninger til de to problemer, så eksisterer der også brugbare løsninger $(x_1, x_2, \dots, x_n)$ og $(y_1, y_2, \dots, y_m)$ med

$$c_1x_1 + c_2x_2 + \dots + c_nx_n = b_1y_1 + b_2y_2 + \dots + b_my_m .$$

Ifølge a) er sådanne $(x_1, x_2, \dots, x_n)$ og $(y_1, y_2, \dots, y_m)$ optimale løsninger, dvs. de to problemer har ens optimale værdier:

$$\min(b_1y_1 + \dots + b_my_m) = \max(c_1x_1 + \dots + c_nx_n)$$

Beviset for a) følger af den teknik, vi anvendte i Eksempel 2.5. Beviset for b) er baseret på en algoritme, som finder de optimale løsninger. Før vi ser nærmere på dette, er det på sin plads at nævne, at Sætning 2.6 er en god sætning. Vi belyser dette i Eksempel 2.6:

<u>Eksempel 2.6</u> Lad det primære og det duale problem være som i Eksempel 2.5. Så er $(u_o, v_o, w_o) = (\frac{6}{5}, \frac{34}{5}, 0)$ en brugbar løsning til det primære problem med værdi

$$30u_o + 35v_o + 20w_o = 30 \cdot \frac{6}{5} + 35 \cdot \frac{34}{5} = 274$$

Er værdien 274 optimal ? Hvis svaret var NEJ, så ville der være en letforklarlig begrundelse, nemlig fremvisningen af en brugbar løsning med en mindre værdi end 274. Imidlertid er svaret JA, og også herfor eksisterer der en letforklarlig begrundelse, nemlig at $(x_o, y_o) = (1, \frac{28}{5})$ er en brugbar løsning til det duale problem med samme værdi:

$$50x_o + 40y_o = 50 + 40 \cdot \frac{28}{5} = 274.$$

Da 274 er en mulig værdi i det duale problem, er 274 en nedre grænse for, hvor langt vi kan nå ned i det primære problem, jvf. Eksempel 2.5, og med (u_o, v_o, w_o) <u>er</u> vi helt dernede, dvs. (u_o, v_o, w_o) <u>er</u> en optimal løsning til det primære problem.

Af argumentationen ovenfor følger også, at $(1, \frac{28}{5})$ er en optimal løsning til det duale problem.

Sætning 2.6 b) sikrer, at det ikke blot er i det tilfældige LP-problem ovenfor, at der for alle svarmuligheder eksisterer en letforklarlig begrundelse, men at det gør det for ethvert LP-problem.

<u>Opgave 2.13</u> Betragt LP-problemet

$$\begin{aligned} \text{max} \quad & 4x_1 + x_2 + 5x_3 + 3x_4 \\ \text{ufa.} \quad & x_1 - x_2 - x_3 + 3x_4 \leqq 1 \\ & 5x_1 + x_2 + 3x_3 + 8x_4 \leqq 55 \\ & -x_1 + 2x_2 + 3x_3 - 5x_4 \leqq 3 \\ & x_1 \geqq 0,\ x_2 \geqq 0,\ x_3 \geqq,\ x_4 \geqq 0 \end{aligned}$$

a) Opstil det duale LP-problem.

b) Vis, at $[x_1 = 0 \;,\; x_2 = 14 \;,\; x_3 = 0 \;,\; x_4 = 5]$ er en brugbar løsning.

c) Find en letforklarlig grund til at løsningen givet i b) er optimal (vink: benyt sætningen om komplementær rest til at finde en brugbar løsning til det duale problem. Benyt så Sætning 2.6 på de to brugbare løsninger).

<u>Opgave 2.14</u> Betragt LP-problemet

$$\begin{aligned} \max \quad & x_1 + x_2 \\ \text{ufa.} \quad & sx_1 + tx_2 \leqq 1 \\ & x_1 \text{ og } x_2 \geqq 0 \end{aligned}$$

a) Opstil det duale LP-problem

b) Find nødvendige og tilstrækkelige betingelser (for s og t) for hvornår 1) der er en optimal løsning, 2) problemet er ubegrænset, og 3) der er ingen brugbare løsninger.

Det faktum, at sætning 2.6 er en god sætning, betyder ikke, at den umiddelbart giver anledning til en metode til at finde den optimale løsning til et LP-problem. Men ligesom beviset for at $\max(\Sigma\alpha_i + \Sigma\beta_j) = \min(\Sigma c_{ij})$ i Sætning 2.5 er baseret på en algoritme, således er også beviset for Sætning 2.6 baseret på en algoritme til at finde den optimale løsning til et LP-problem. Algoritmen kaldes *simplex-algoritmen*, og den blev udviklet af den amerikanske matematiker G.B. Dantzig i 1947.

Vi skal ikke her give en detaljeret gennemgang af simplex-algoritmen, men blot se på, hvordan det duale problem i Eksempel 2.5 kan løses vha. algoritmen.

<u>Eksempel 2.7</u>

$$\begin{aligned} \text{Find} \quad \max \quad & 50x + 40y = z \\ \text{ufa.} \quad & 2x + 5y \leqq 30 \\ & 7x + 5y \leqq 35 \\ & 5x + 2y \leqq 20 \\ & x \geqq 0 \;,\; y \geqq 0. \end{aligned}$$

Første skridt er at omforme ulighederne til ligheder. Dette sker ved indførelse af nye såkaldte *rest-variable*. Det er let at se, at følgende problem har samme løsning som det oprindelige:

$$
\begin{array}{llcl}
\text{Find max} & 50x + 40y & = & z \\
\text{ufa.} & 2x + 5y + q & = & 30 \\
 & 7x + 5y + s & = & 35 \\
 & 5x + 2y + t & = & 20 \\
 & x \geqq 0 \,,\ y \geqq 0 \,,\ q \geqq 0 \,,\ s \geqq 0 \,,\ t \geqq 0 & &
\end{array}
$$

Vi har altså fået ligheder nu, og det er en fordel, da det er lettere at løse ligningssystemer end ulighedssystemer. Prisen er flere variable. Begrænsningerne i vort problem består nu af 3 ligninger med 5 ubekendte, som alle skal være $\geqq 0$.

Strategien i simplex-algoritmen er, at bevæge sig fra løsning til løsning, således at z-værdien hele tiden forbedres, indtil dette ikke længere er muligt.

Vi starter med den oplagte løsning.

(I) $(x,y,q,s,t) = (0,0,30,35,20)$,

hvor $z = 50x + 40y = 0$

Vi kan finde en bedre løsning ved at forøge enten x eller y, og samtidig formindske q, s og t.

Lad os vælge at forøge x. Den første ligning viser, at vi kan forøge x til højst 15, da q så når ned på 0. Den anden ligning viser tilsvarende, at vi kan forøge x til højst 5. Endelig viser den tredie ligning, at vi kan forøge x til højst 4. Alt i alt kan vi derfor kun forøge x til 4. Gør vi det, formindskes q til 22 p.gr.af ligning 1, s formindskes til 7 p.gr.af ligning 2, og t formindskes til 0 p.gr.af ligning 3. Som den næste løsning får vi altså $(x,y,q,s,t) = (4,0,22,7,0)$. Denne løsning fremgår tydeligere (som (I) fremgik af det oprindelige ligningssystem) ved at indsætte

$$x = 4 - \frac{2}{5}y - \frac{1}{5}t \,,$$

opnået fra den begrænsende ligning, dvs. den tredie, i de to øvrige ligninger. Derved fås ligningssystemet på formen:

$$
\begin{aligned}
\tfrac{21}{5}y + q \quad - \tfrac{2}{5}t &= 22 \\
\tfrac{11}{5}y \quad + s - \tfrac{7}{5}t &= 7 \\
x + \tfrac{2}{5}y \quad + \tfrac{1}{5}t &= 4
\end{aligned}
$$

og løsningen fremgår umiddelbart ved at sætte $y = t = 0$:

(II) $(x,y,q,s,t) = (4,0,22,7,0)$, hvor

$$z = 50x + 40y = 200 + 20y - 10t = 200$$

hvor vi også i udtrykket for z har indsat x, således at z igen er udtrykt ved de to variable, som i øjeblikket er 0.

Situationen er nu helt analog til situationen, da vi havde løsningen (I). Da koefficienten til t i z er negativ, kan det ikke betale sig at forøge t. Vi vil derfor forøge y. Som før kan y maximalt forøges med

$$\min(22/\tfrac{21}{5}\ ,\ 7/\tfrac{11}{5}\ ,\ 4/\tfrac{2}{5}) = \tfrac{35}{11}$$

og det er den anden ligning, som sætter begrænsningen. Ved omformning af ligningssystemet som før fås

$$
\begin{aligned}
q - \tfrac{21}{11}s + \tfrac{25}{11}t &= \tfrac{95}{11} \\
y \quad + \tfrac{5}{11}s - \tfrac{7}{11}t &= \tfrac{35}{11} \\
x \quad - \tfrac{2}{11}s + \tfrac{5}{11}t &= \tfrac{30}{11}
\end{aligned}
$$

og den næste løsning bliver

(III) $(x,y,q,s,t) = (\tfrac{30}{11}\ ,\ \tfrac{35}{11}\ ,\ \tfrac{95}{11}\ ,\ 0\ ,\ 0)$, hvor

$$z = \tfrac{2900}{11} - \tfrac{100}{11}s + \tfrac{30}{11}t = 263 + \tfrac{7}{11}$$

Ved forøgelse af t, er det nu den første ligning, som sætter begrænsningen. Med samme teknik som ovenfor fås ligningssystemet

$$\begin{aligned} \tfrac{11}{25}q - \tfrac{21}{25}s + t &= \tfrac{19}{5} \\ y + \tfrac{7}{25}q - \tfrac{2}{25}s &= \tfrac{28}{5} \\ x - \tfrac{1}{5}q + \tfrac{1}{5}s &= 1 \end{aligned}$$

og løsningen

(IV) $(x,y,q,s,t) = (1, \frac{28}{5}, 0, 0, \frac{19}{5})$, hvor

$$z = 274 - \tfrac{6}{5}q - \tfrac{34}{5}s = 274$$

Af den form z her har fået, fremgår det umiddelbart, at max $z = 274$, og at dette opnås, når $q = s = 0$. Altså er problemet løst.

De to koefficienter til q og s i det afsluttende udtryk er $\leqq 0$. Disse tal numerisk er netop optimale værdier for u og v i det primære problem, jvf. Eksempel 2.6. Dette er ikke tilfældigt! Når simplex-algoritmen standser, kan man af slut-situationen direkte aflæse brugbare løsninger til såvel det primære som det duale problem, og de to løsninger giver samme værdi i de to problemer. Dette er præcis idéen i et bevis for Sætning 2.6 b).

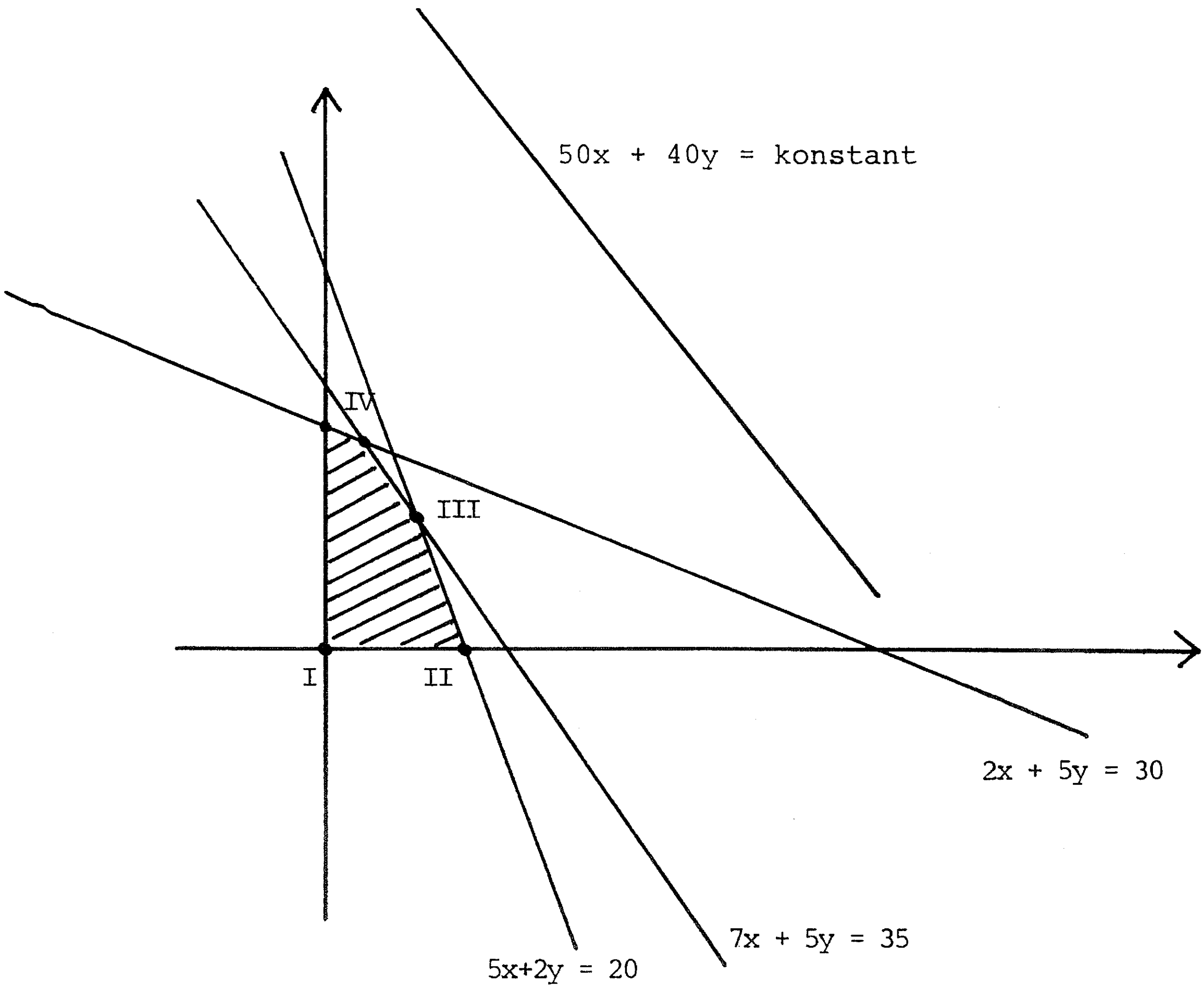
En alternativ måde at finde de optimale duale variable på ud fra $(1, \frac{28}{5}, 0, 0, \frac{19}{5})$, er at udnytte sætningen om komplementær rest, således som det skete i Opgave 2.13. Prøv det !

↓ Den form, som z har fået i den afsluttende løsning

$$z = 274 - \tfrac{6}{5}q - \tfrac{34}{5}s$$

viser som sagt, at $q = s = 0$ giver optimalitet, dvs. der gælder lighedstegn for optimale x og y i den første og anden ulighed i det problem, som vi startede dette eksempel med. Fortolkes $z = 50x + 40y$ som en fortjeneste, der skal maximeres, så ser vi, at hvis f.eks. tallet 30 i den ulighed reduceres til $30-\varepsilon$, så må vi øge q fra 0 til ε, og fortjenesten daler så med $\frac{6}{5}\varepsilon$. Tallet $\frac{6}{5}$, som er værdien af den optimale variabel u_o i det duale problem, viser altså hvordan ændringer i tallet 30 påvirker den maximale fortjeneste. I en sådan økonomisk fortolkning af LP-problemet, kaldes optimale duale variable derfor for *skyggepriser*. ↑

Det problem, som vi løste (med simplex-algoritmen), har kun 2 variable x og y, og det kan derfor også løses rent geometrisk. Hvordan antydes på følgende figur, hvor de brugbare løsninger er de (x,y), som ligger i det skraverede område:



Det ses, at de fire løsninger, vi betragtede i simplex-algoritmen, svarer til fire hjørner i det brugbare område. Det er generelt sådan: simplex-algoritmen består i at bevæge sig fra hjørne til hjørne i det brugbare område, indtil man finder et optimalt hjørne (i dette tilfælde IV).

Opgave 2.15 Find vha. simplex-algoritmen

$$\begin{array}{ll} \max & 5x_1 + 4x_2 + 3x_3 \\ \text{ufa.} & 2x_1 + 3x_2 + x_3 \leqq 5 \\ & 4x_1 + x_2 + 2x_3 \leqq 11 \\ & 3x_1 + 4x_2 + 2x_3 \leqq 8 \\ & x_1 \geqq 0 \ , \ x_2 \geqq 0 \ , \ x_3 \geqq 0 . \end{array}$$

Er simplex-algoritmen en god algoritme ? Desværre er det ikke engang helt klart, at den er en algoritme, som altid standser! Man kan konstruere tilfælde, hvor den vender tilbage til en tidligere løsning og derfor kommer til at cykle uendeligt. Dette kan undgås ved at fastsætte bestemte regler for 1) hvilken variabel, man skal vælge som den næste at forøge, og 2) hvilken begrænsende liging man skal vælge i de tilfælde, hvor to eller flere ligninger begrænser samtidig.

Man kan altså præcisere regler for simplex-algoritmen, så den slutter, og dette kan gøres på flere forskellige måder. Men desværre kan man for nogle af de forskellige muligheder, der findes, konstruere eksempler, hvor simplex-algoritmen bruger et meget stort antal skridt. <u>Det er et uløst problem, om man kan fastsætte reglerne i simplex-algoritmen således, at den bliver en god algoritme i den tekniske forstand, vi har defineret i §I.1</u>.

I praksis har simplex-algoritmen dog vist sig meget effektiv. For problemer med n variable, og m begrænsninger synes kompleksiteten for næsten alle tilfælde at være højst $0(n \cdot m)$. Men der kan altså desværre konstrueres specielle eksempler, hvor antal skridt ikke engang er begrænset af noget polynomium i n og m.

Situationen er altså utilfredsstillende. Simplex-algoritmen er i praksis en meget effektiv algoritme, som benyttes meget i forbindelse med løsning af mange praktiske problemer. Men den er <u>ikke</u> god i den forstand, vi har defineret. Det tyder på, at vores definition måske ikke er helt tilfredsstillende.

Desuden synes det, som om vi har fundet et tilfælde, der ikke lever op til filosofien om, at gode sætninger og gode algoritmer følges ad.

Lad os til slut vende tilbage til job-tilordning. Dette problem kan beskrives således:

<u>Job-tilordning</u>

Find min $\Sigma\, c_{ij}\, x_{ij}$

$$\text{ufa.} \quad \sum_{i=1}^{n} x_{ij} = 1 \quad \text{for alle} \quad j = 1,2,\ldots,n$$

$$\sum_{j=1}^{n} x_{ij} = 1 \quad \text{for alle} \quad i = 1,2,\ldots,n$$

$$x_{ij} = 0 \quad \text{eller} \quad 1 \quad \text{for alle} \quad i \quad \text{og} \quad j$$

Dette er <u>ikke</u> et LP-problem, idet x_{ij} er forudsat kun at have de heltallige værdier 0 og 1. Erstatter vi imidlertid denne begrænsning med $x_{ij} \geqq 0$ fås en udvidelse af det brugbare område, og vi må på forhånd forvente at kunne komme længere ned med minimum. Problemet er så et LP-problem:

<u>Primære LP-job-tilordning</u>

Find min $\Sigma\, c_{ij}\, x_{ij}$

$$\text{ufa.} \quad \sum_{i=1}^{n} x_{ij} = 1 \quad \text{for alle} \quad j=1,2,\ldots,n$$

$$\sum_{j=1}^{n} x_{ij} = 1 \quad \text{for alle} \quad i=1,2,\ldots,n$$

$$x_{ij} \geqq 0 \quad \text{for alle} \quad i,j.$$

Dette problem har det duale problem

Duale LP-job-tilordning

Find max $(\sum_i \alpha_i + \sum_j \beta_j)$

ufa. $\alpha_i + \beta_j \leqq c_{ij}$ for alle i,j

α_i , β_j ubegrænsede.

Sætningerne 2.5 og 2.6 a) viser, at den optimale løsning i det primære problem kan opnås med x_{ij}'er, som er enten 0 eller 1. Så selv om vi udvidede det brugbare område, da vi gik fra job-tilordning til LP-job-tilordning, så er minimum det samme i de to problemer. Dette er overraskende, for LP-problemer generelt har <u>ikke</u> heltallige optimale løsninger, heller ikke selv om der kun indgår heltal i formuleringen, jvf. Eksempel 2.7.

Vi har vha. den gode Sætning 2.5 set, at job-tilordning kan formuleres som et LP-problem, og Sætning 2.5 kan så opfattes som et special-tilfælde af LP-dualitet. Noget tilsvarende gør sig gældende for Sætningerne 2.1 og 2.3.

II.5 TRANSPORT-PROBLEMET.

I *transport-problemet* er udgangspunktet en n×m-matrix med tal c_{ij} i række i og søjle j. Rækkerne svarer til n lagre $A_1, A_2, \dots, A_n$ med $a_1, a_2, \dots, a_n$ enheder af en vare V. Søjlerne svarer til m kunder $B_1, B_2, \dots, B_m$ med behov for $b_1, b_2, \dots, b_m$ enheder af V. Det forudsættes at $\Sigma a_i = \Sigma b_j$.

Problemet er at angive en måde at transportere de Σa_i enheder fra lagrene til kunderne, således at x_{ij} enheder går fra A_i til B_j , og således at de samlede transport-omkostninger $\Sigma c_{ij}x_{ij}$ bliver så lave som muligt.

Der er tale om en speciel type af LP-problem:

Transport-problemet

Find min $\Sigma c_{ij}x_{ij}$

ufa. $\sum_{j=1}^{n} x_{ij} = a_i$ for alle $i=1,2,\dots,m$

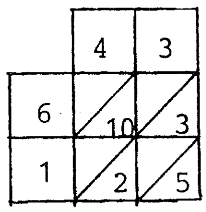
$\sum_{i=1}^{m} x_{ij} = b_j$ for alle $j=1,2,\dots,n$

$x_{ij} \geqq 0$ for alle i,j.

Tilfældet m=n og $a_i = b_j = 1$ for alle i og j er job-tilordnings-problemet, som altså er et specielt tilfælde. Transport-problemet kan løses vha. simplex-algoritmen, men ligesom det var tilfældet for job-tilordning, er der en mere effektiv algoritme, som udnytter den specielle struktur.

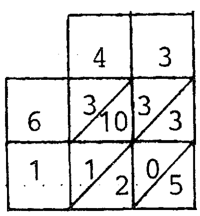
Transport-problemet er en meget anvendelig matematisk model. Man kan f.eks. tænke på varen V som sand, der fra en række depoter skal køres ud til byggepladser. Eller V kunne være busser, som om morgenen skal køre fra garageanlæg rundt i byen ud til alle endestationerne. I det første tilfælde er x_{ij}'erne ikke nødvendigvis heltallige, men det er de i det andet tilfælde. Vi skal se, at hvis a_i'erne og b_j'erne alle er heltallige, så eksisterer der altid en optimal heltalsløsning til transport-problemet i LP-formuleringen ovenfor. Transport-problemet kaldes også ofte i den matematiske litteratur for *Hitchcock-problemet* efter den amerikanske matematiker F.L. Hitchcock, som studerede det i 1941. Den russiske økonom L.V. Kantorovich havde dog allerede i 1939 benyttet en løsning af transport-problemet i transport-planlægning, men dette forblev længe upåagtet, både indenfor og udenfor USSR (men i 1975 fik Kantorovich samen med den amerikanske økonom T.C. Koopmans Nobel-prisen i økonomi for deres bidrag til teorien for optimal placering af ressourcer).

Eksempel 2.8 Betragt transportproblemet

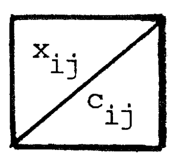


hvor $c_{11} = 10$, $c_{12} = 3$, $c_{21} = 2$, $c_{22} = 5$, $a_1 = 6$, $a_2 = 1$, $b_1 = 4$ og $b_2 = 3$.

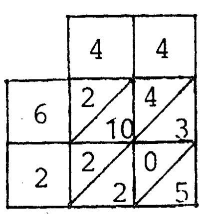
En brugbar løsning med $\Sigma c_{ij} x_{ij} = 41$ er :



Felt (i,j) er udfyldt således:



Det kan vises, at denne løsning er optimal. Ved at forøge a_2 og b_2 hver med 1 fås en brugbar løsning med $\Sigma c_{ij}\ x_{ij} = 36$:



Altså har vi ved at sende en større mængde opnået en billigere løsning!

Opgave 2.16. Betragt et transportproblem med $m = n = 2$. Benyt strategien at sende mest muligt fra A_i til B_j med det mindste c_{ij}. Send dernæst mest muligt fra A_i til B_j med det næstmindste c_{ij}, osv.

a) Vis, at den beskrevne strategi ikke altid giver en optimal løsning.

b) Antag, at c_{11} er den mindste af de fire c_{ij}'er. Vis, at strategien virker hvis og kun hvis

$$c_{11} + c_{22} \leqq c_{21} + c_{12}$$

c) Kan du finde en forbedret strategi, som altid giver en optimal løsning?

Med den erfaring, vi har opnået i §§II.3 og II.4 er det ikke svært at opstille en løsningsmodel for transport-problemet. Idéen er igen at fratrække tal α_i fra rækkerne og tal β_j fra søjlerne i c_{ij}-matricen. Derved ændres omkostningerne, men det fine er, at en brugbar løsning x_{ij} er optimal for det nye problem hvis og kun hvis den er for det oprindelige. Strategien er nu, at fratrække tal α_i og β_j, så alle omkostninger i det nye problem er $\geqq 0$, og så vi kan sende alle enheder af V af sted ved kun at bruge forbindelser fra A_i til B_j med ny omkostning 0. Dvs. α_i'erne og β_j'erne skal opfylde:

(*) $\alpha_i + \beta_j \leqq c_{ij}$ for alle i og j

(**) Der eksisterer par (i,j) med $\alpha_i + \beta_j = c_{ij}$, således at alle enheder af V kan sendes ved kun at sende fra A_i'er til B_j'er svarende til disse par.

For rækkevægte α_i og søjlevægte β_j som opfylder (*), og for en brugbar løsning x_{ij}, gælder

(***) $$\sum_i a_i\alpha_i + \sum_j b_j\beta_j \leqq \Sigma c_{ij}\, x_{ij}$$

Bevis: $\sum_{i,j} c_{ij}\, x_{ij} \geqq$

$\sum_{i,j} (\alpha_i + \beta_j) x_{ij} =$

$\sum_i \sum_j \alpha_i x_{ij} + \sum_j \sum_i \beta_j x_{ij} =$

$\sum_i \alpha_i a_i + \sum_j \beta_j b_j$ ■

Lykkes (**), gælder der lighedstegn i (***) for de pågældende α_i'er , β_j'er og x_{ij}'er. Analysen ovenfor viser, at sådanne x_{ij}'er har $\Sigma c_{ij} x_{ij}$ mindst mulig. Dette følger også af lighedstegnet i (***), idet vi så med $\Sigma c_{ij} x_{ij}$ er helt nede på den nedre grænse $\Sigma a_i \alpha_i + \sum_j b_j \beta_j$. (***) viser samtidig, at de pågældende α_i'er og β_j'er har $\Sigma a_i \alpha_i + \Sigma b_j \beta_j$ størst mulig.

At (**) altid kan lykkes, følger umiddelbart af LP-dualitets-sætningen Sætning 2.6, idet det duale problem til transport-problemet ser således ud:

Duale transport-problem

Find max $\sum_i a_i \alpha_i + \sum_j b_j \beta_j$

ufa. $\alpha_i + \beta_j \leqq c_{ij}$ for alle i,j

α_i , β_j ubegrænsede.

Et andet argument for at (**) altid kan lykkes, er at angive en direkte algoritme! Vi tager et par eksempler først.

Eksempel 2.9. Transportproblemet

	4	3
6	10	3
1	2	5

har som angivet i Eksempel 2.8 en brugbar løsning x_{ij} med $\Sigma c_{ij}x_{ij} = 41$. At denne løsning er optimal, har en letforklarlig begrundelse, nemlig at vægtene $\alpha_1 = 0$, $\alpha_2 = -8$, $\beta_1 = 10$ og $\beta_2 = 3$ opfylder (*) og dermed (***), og at $\Sigma a_i\alpha_i + \Sigma b_j\beta_j = 41$, dvs. samme sum som $\Sigma c_{ij}x_{ij}$.

De angivne vægte giver også noget af en forklaring på paradoxet i Eksempel 2.8. Da $\alpha_2 + \beta_2$ er negativ, bliver $\Sigma a_i\alpha_i + \Sigma b_i\beta_j$ mindre, hvis vi øger a_2 og b_2 med samme tilvækst. Kan dette klares, så transporten stadig sker fra A_i'er til B_j'er, hvor $\alpha_i + \beta_j = c_{ij}$, så fås en billigere transport af en større mængde.

Eksempel 2.10 Transportproblemet

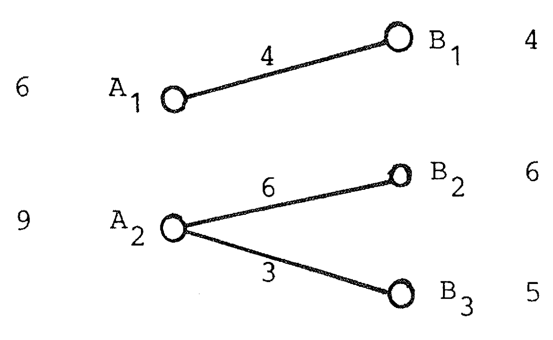
	4	6	5
6	3	5	7
9	6	4	3

vil vi nu søge at løse vha. den foreslåede teknik.

Vi starter med rækkevægtene $\alpha_1 = \min(c_{1j}) = 3$ og $\alpha_2 = \min(c_{2j}) = 3$. Herefter er de bedst mulige søjlevægte $\beta_1 = 0$, $\beta_2 = 1$ og $\beta_3 = 0$:

		0	1	0
		4	6	5
3	6	3	5	7
3	9	6	4	3

Herefter sender vi mest muligt af V af sted ved kun at benytte forbindelserne med ○ , dvs. forbindelserne med $\alpha_i + \beta_j = c_{ij}$:



		0	1	0
		4	6	5
3	6	3	5	7
3	9	6	4	3

Ved A_1 er der mere af V til rådighed. Dette kan sendes til B_1, men B_1 ønsker ikke mere. Ved A_2 er der ikke mere af V til rådighed. De punkter, hvorfra eller hvortil mere kan sendes fra A_i'erne langs de tilladte forbindelser, er således A_1 og B_1. Disse punkter mærkes.

Fra de mærkede A_i'er til de ikke-mærkede B_j'er, er der ingen tilladte forbindelser. Vi forhøjer nu α_i'erne svarende til de mærkede A_i'er og formindsker tilsvarende β_j'erne svarende til de mærkede B_j'er, så meget som muligt. I det foreliggende tilfælde kan α_1 forhøjes med højst 1 (p.gr.af $c_{12} = 5$) og β_1 formindskes med det samme beløb:

		* -1	1	0
		4	6	5
* 4	6	4 / 3	5	7
3	9	6	6 / 4	3 / 3

Nu kan mærkningen fortsætte, idet der kan sendes fra A_1 til B_2. Derfor mærkes B_2, men B_2 ønsker ikke mere. Imidlertid bliver der plads til at sende noget fra A_1 til B_2, hvis strømmen fra A_2 til B_2 formindskes. Og så bliver der plads til at sende noget fra A_2, som derfor mærkes. Dette kan sendes til B_3, som også mærkes. Og B_3 ønsker mere!

Derfor kan strømmen øges ved at sende fra A_1 til B_2 (højst 2, da der kun er 6 ialt til rådighed ved A_1), formindske strømmen fra A_2 til B_2 (med højst 6, da der kun går 6 fra A_2 til B_2), og sende fra A_2 til B_3 (højst 2, da B_3 højst skal have 5). Alt i alt kan strømmen derfor øges med 2:

		-1	1	0
		4	6	5
4	6	4 / 3	2 / 5	7
3	9	6	4 / 4	5 / 3

Hermed er alle enheder af V sendt og modtaget, og der er kun benyttet forbindelser (A_i, B_j), hvor $\alpha_i + \beta_j = c_{ij}$.

Den angivne strøm er derfor en ønsket billigste strøm. Vi ser at

$$\Sigma c_{ij} x_{ij} = 4 \cdot 3 + 2 \cdot 5 + 4 \cdot 4 + 5 \cdot 3 = 53$$

og

$$\Sigma a_i \alpha_i + \Sigma b_j \beta_j = 6 \cdot 4 + 9 \cdot 3 + 4 \cdot (-1) + 6 \cdot 1 + 5 \cdot 0 = 53.$$

De angivne α_i'er og β_j'er er en letforklarlig begrundelse for at $\Sigma c_{ij} x_{ij}$ har minimum 53.

Ingredienserne i Eksempel 2.10 er præcis de, som indgår generelt. Vi ser, at algoritmen helt kommer til at svare til ALGORIME JOB-TILORDNING, blot erstattes mærkningen i ALGORITME PARDANNELSE af en lidt ændret mærkningsprocedure:

Først mærkes alle A_i, hvorfra der ikke er sendt a_i enheder. Så mærkes punkterne i $\{A_1, A_2, \ldots, A_n\}$ og $\{B_1, B_2, \ldots, B_m\}$ skiftevis efter reglerne

a) for alle ny-mærkede A_i mærkes alle ikke-mærkede B_j forbundet til A_i med kant i G ,

b) for alle ny-mærkede B_j mærkes alle ikke-mærkede A_i forbundet med kant til B_j, hvor strømmen x_{ij} fra A_i til B_j er > 0.

Får man ved denne proces mærket et B_j, hvor den samlede strøm til B_j er $< b_j$, så kan strømmen forøges langs den fundne "alternerende" vej. Får man ikke mærket et sådant B_j, så kan α_i'erne og β_j'erne forbedres præcis som beskrevet i ALGORITME JOB-TILORDNING, hvor $\mathcal{A}$ igen er de mærkede A_i'er og $\mathcal{B}$ de mærkede B_j'er.

↓ <u>Opgave 2.17</u> Giv en formulering af ALGORITME TRANSPORT i stil med ALGORITME JOB-TILORDNING.

Vink: I ALGORITME JOB-TILORDNING benyttedes som hjælpealgoritme ALGORITME PARDANNELSE. Tilsvarende kan i ALGORITME TRANSPORT benyttes en hjælpealgoritme ALGORITME STRØM, som vha. mærkningen ovenfor finder en strøm x_{ij}, fra A_i til B_j for alle i og j , hvor

1) der er ingen kant fra et mærket A_i til et ikke-mærket B_j ,

2) der er ingen strøm, dvs. $x_{ij} = 0$, fra et ikke-mærket A_i til et mærket B_j ,

3) de mærkede B_j'er er alle helt imødekommet, dvs. $\sum_i x_{ij} = b_j$.

† En sådan strøm kan vises at være en størst mulig strøm.

Metoden skitseret ovenfor til løsning af transport-problemet ligner ALGORITME JOB-TILORDNING så meget, at den også kaldes *den ungarske metode*. Af og til benyttes også navnet α-β-*metoden*.

Lad os til slut formulere den tilsvarende gode sætning

<u>Sætning 2.7</u> For transport-problemet gælder at

$$\min \sum_{i,j} c_{ij} x_{ij} = \max(\sum_i a_i \alpha_i + \sum_j b_j \beta_j)$$

hvor $\alpha_i + \beta_j \leqq c_{ij}$ for alle i og j (og hvor α_i'erne og β_j'erne ellers er ubegrænsede).

Sætning 2.7 er blot et special-tilfælde af LP-dualitetssætningen Sætning 2.6. Et mere direkte bevis giver algoritmen for transport-problemet, som netop slutter med x_{ij}'er, α_i'er og β_j'er, som opfylder sætningen. Denne udgave af beviset er mere tilfredsstillende, idet den også viser:

<u>Sætning 2.8</u> a) For et transport-problem med heltallige a_i'er og b_j'er, eksisterer der en optimal løsning til problemet med alle x_{ij}'er heltallige.

b) For et dualt transport-problem med heltallige c_{ij}'er eksisterer der en optimal løsning til problemet med alle α_i'er og β_j'er heltallige.

Opgave 2.18 Løs transport-problemet givet ved

		B_1	B_2	B_3	B_4
		5	3	5	2
A_1	5	5	4	7	6
A_2	4	2	5	3	2
A_3	6	6	3	4	4

Angiv en optimal dual løsning, og giv en letforklarlig grund til at din løsning på transport-problemet virkelig er optimal (så din lærer ikke behøver at kontrollere, hvordan du fandt løsningen!).

II.6 LITTERATUR.

Pardannelse er beskrevet i alle bøger om grafteori, også naturligvis i König's fremragende og grundige:

D. König: Theorie der endlichen und unendlichen Graphen , Leipzig 1936, genoptrykt af Chelsea Publ. Comp. New York 1950.

Pardannelse for ikke-todelte grafer blev først studeret af danskeren J. Petersen i 1891. Han er i dag et kendt navn over hele verden på grund af denne pionér-indsats. Se König's bog.

Pardannelse i det vægtede to-delte tilfælde behandles i den af Bondy & Murty i §I.7 nævnte bog.

Pardannelse for ikke to-delte grafer (ikke vægtet og vægtet) blev løst generelt med gode sætninger og gode algoritmer af den canadiske matematiker J. Edmonds i 1960'erne, se litteratur-henvisningen i §I.7. En god sætning for det ikke-vægtede tilfælde var dog fundet allerede i 1947 af den canadiske matematiker W.T. Tutte, se Bondy & Murty's bog.

Biblen om LP er skrevet af den amerikanske matematiker, som især bidrog til udviklingen af LP i årene efter krigen:

G.B. Dantzig: Linear programming and extensions , Princeton Univ. Press 1963.

Der er ellers et hav af LP-bøger. Den mest fremragende, både indholdsmæssigt og pædagogisk, er

V. Chvátal: Linear programming , W.H. Freeman 1983.

På dansk foreligger

M.Blomhøj, K.Frisdahl & F.Mølgaard Olsen : Lineær programmering, FAG 1984.

Mange forskellige planlægningsproblemer, incl. job-tilordning og transport, er behandlet i små korte kapitler i

S. Vajda: Planning by mathematics , Pitman 1973.

The Open University har et kursus i "Graphs, Networks and Designs" og der er i denne forbindelse udgivet hæfter, bånd og video-film. Et af hæfterne er

Assignment and transportation, The Open University 1981.

Strømning i netværk generelt, herunder transport-problemet, er udførligt beskrevet i den af Ford og Fulkerson i §I.7 nævnte bog.

Mange af opgaverne og eksemplerne i dette kapitel er hentet i ovennævnte kilder.