

# Distribueringsproblemer og løsningsmetoder – en oversigt

Af JAKOB KRARUP<sup>1)</sup> og PETER MARK PRUZAN<sup>2)</sup>

## 1. *Introduktion.*

Formålet med denne artikel er at give en oversigt over en række typiske problemstillinger, der for de flestes vedkommende er udførligt behandlet i litteraturen, og som alle har en vis – større eller mindre – relation til dimensionering og drift af distributionssystemer.

Lineær programmering og Simplex-proceduren anses med rette for at være et af operationsanalytikerens mest slagkraftige hjælpemidler; en dybtgående gennemgang af teorien samt en omfattende litteraturliste kan f. eks. findes hos G. B. Dantzig (1963). Principielt besidder flere af de her betragtede problemstillinger sådanne egenskaber, der muliggør en anvendelse af Simplex-proceduren, men – som vi skal se – er dette ofte både ubekvem i formuleringen og uøkonomisk ved numerisk løsning på datamaskiner. Den indirekte årsag hertil er distribueringsproblemers ofte udprægede »netværksstruktur«, hvilket har foranlediget, at teorien for flow i netværk har udskilt sig som en selvstændig disciplin under matematisk programmering i almindelighed og – i første omgang – under lineær programmering i særdeleshed. Det uoversættelige ord »flow« omfatter simpelthen »alt det, der flyder« (varer, lastvogne, penge, informationer etc. etc.) i det net, der indgår i modellen for et forelagt optimeringsproblem.

Vi har søgt at klassificere de problemer, der i de følgende afsnit skal studeres nøjere, efter udseendet af det til hvert enkelt problem svarende

<sup>1)</sup> Lektor, lic. techn., Institut for Matematisk Statistik og Operationsanalyse, Danmarks tekniske Højskole.

<sup>2)</sup> Dr., Operations Analysis Corp. A/S og lektor, Institut for Matematisk Statistik og Operationsanalyse, Danmarks tekniske Højskole.

# Distribueringsproblemer og løsningsmetoder – en oversigt

Af JAKOB KRARUP<sup>1)</sup> og PETER MARK PRUZAN<sup>2)</sup>

## 1. *Introduktion.*

Formålet med denne artikel er at give en oversigt over en række typiske problemstillinger, der for de flestes vedkommende er udførligt behandlet i litteraturen, og som alle har en vis – større eller mindre – relation til dimensionering og drift af distributionssystemer.

Lineær programmering og Simplex-proceduren anses med rette for at være et af operationsanalytikerens mest slagkraftige hjælpemidler; en dybtgående gennemgang af teorien samt en omfattende litteraturliste kan f. eks. findes hos G. B. Dantzig (1963). Principielt besidder flere af de her betragtede problemstillinger sådanne egenskaber, der muliggør en anvendelse af Simplex-proceduren, men – som vi skal se – er dette ofte både ubekvemt i formuleringen og uøkonomisk ved numerisk løsning på datamaskiner. Den indirekte årsag hertil er distribueringsproblemers ofte udprægede »netværksstruktur«, hvilket har foranlediget, at teorien for flow i netværk har udskilt sig som en selvstændig disciplin under matematisk programmering i almindelighed og – i første omgang – under lineær programmering i særdeleshed. Det uoversættelige ord »flow« omfatter simpelthen »alt det, der flyder« (varer, lastvogne, penge, informationer etc. etc.) i det net, der indgår i modellen for et forelagt optimeringsproblem.

Vi har søgt at klassificere de problemer, der i de følgende afsnit skal studeres nøjere, efter udseendet af det til hvert enkelt problem svarende

<sup>1)</sup> Lektor, lic. techn., Institut for Matematisk Statistik og Operationsanalyse, Danmarks tekniske Højskole.

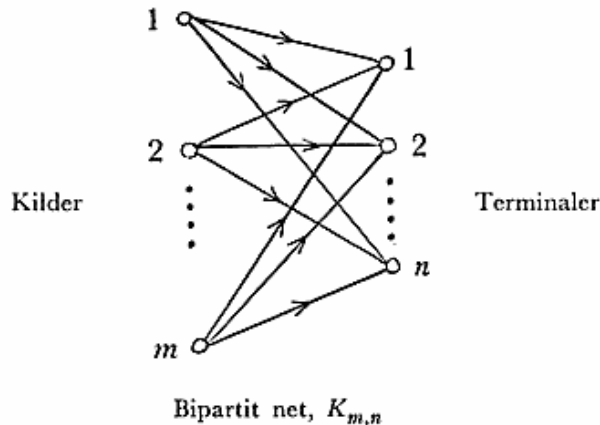
<sup>2)</sup> Dr., Operations Analysis Corp. A/S og lektor, Institut for Matematisk Statistik og Operationsanalyse, Danmarks tekniske Højskole.

net. I fremstillingen har vi så vidt muligt undgået brug af matematiske symboler og grafteoretiske begreber. Der er heller ikke redegjort i detaljer for det matematiske indhold i de algoritmer, der kommer på tale, og interessererede henvises i stedet til litteraturfortegnelsen, der findes samlet til slut. Alle personnavne i teksten refererer ligeledes til denne fortegnelse.

Klassifikationen bygger på følgende intuitivt forståelige begreber:

Et orienteret netværk eller blot et *orienteret net* er sammensat af *punkter* og *grene*, hvor hver gren udgår fra et *startpunkt* og ender i et *slutpunkt*. Flowbegrebet henføres til grenene, og vi taler om (statisk) flow i gren  $(x,y)$  som den mængde flow, der pr. tidsenhed flyder i denne gren fra  $x$  til  $y$ . Punkterne i et net kan være enten *kilder*, *mellemliggende punkter* eller *terminaler*. For et mellemliggende punkt er den flowmængde, der løber til punktet, lig den flowmængde, der udgår fra punktet; sammenlign elektriske net. For kilder kan den udgående flowmængde overstige den indgående og omvendt for terminaler. Flow kan med andre ord *genereres* i kilder og *absorberes* ved terminaler.

Et net kaldes *bipartit*, hvis det ikke indeholder mellemliggende punkter, og hvis der fra enhver kilde findes en gren til enhver terminal. Et bipartit net med  $m$  kilder og  $n$  terminaler har således  $m \times n$  grene og betegnes i grafteorien med symbolet  $K_{m,n}$ :



I afsnittene 2–6 diskuteres problemer, der alle kan henføres til  $K_{m,n}$ . Dernæst følger i afsnittene 7–9 forskellige situationer med relation til vilkårlige, orienterede net og endelig (afsn. 10–13) nogle outsiders, hvis placering i en af de to nævnte grupper er diskutabel. Inden for hver hovedgruppe foretages den videre opdeling på grundlag af forskellige egenskaber (eller kombinationer heraf), der kan tillægges nettets punkter og grene.

Som konklusion vises i afsnit 14 et skema over samtlige her omtalte problemer stillet over for seks »familier« af relevante løsningsmetoder.

## 2. Det klassiske transportproblem.

Lad der være givet  $m$  producenter og  $n$  forbrugere af en enkelt varetype. Den øvre grænse for hver enkelt fabriks produktion pr. tidsenhed samt den nedre grænse for hver enkelt forbrugers efterspørgsel pr. tidsenhed er kendt. Der findes direkte transportveje fra enhver fabrik til enhver forbruger, og til hver af de ialt  $m \times n$  transportveje er knyttet en given enhedspris. Bestem en varefordeling i nettet, der imødekommer kravene om kapacitet og efterspørgsel og som minimaliserer de totale transportomkostninger.

Vi indfører betegnelserne:

- $a_i$  : Maximal produktion for fabrik  $i$ ,  $i=1, \dots, m$
- $b_j$  : Efterspørgsel for forbruger  $j$ ,  $j=1, \dots, n$
- $c_{ij}$  : Enhedstransportpris, fabrik  $i \rightarrow$  forbruger  $j$ .
- $x_{ij}$  : Antal vareenheder, fabrik  $i \rightarrow$  forbruger  $j$ .

og kan formulere problemet:

Bestem de  $m \times n$  værdier,  $x_{ij}$ ,

der minimerer 
$$\sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij}$$

under begrænsningerne

$$\sum_{j=1}^n x_{ij} \leq a_i; \quad i=1, \dots, m$$

$$\sum_{i=1}^m x_{ij} \geq b_j; \quad j=1, \dots, n$$

$$x_{ij} \geq 0;$$

Det ses umiddelbart, at en både nødvendig og tilstrækkelig betingelse for eksistensen af en løsning er, at den totale produktion er større end eller lig den totale efterspørgsel. Hvis der produceres mere end der efterspørges, tilføjes blot en ekstra forbruger, til hvem transport fra enhver fabrik er gratis, og hvis efterspørgsel er lig den overskydende produktion.

Det underliggende net er atter her  $K_{m,n}$  ( $m$  kilder og  $n$  terminaler), og vi er begribeligvis ikke interesserede i ved løsningen at finde en optimal flowfordeling i nettet, hvor  $2/7$  vareenhed flyder i én gren og  $5/8$  i en anden, dersom »vareenheder« udgøres af udelelige objekter. Vi bør derfor tilføje den ekstra restriktion, at alle positive  $x_{ij}$  skal være heltal, men dette

kompliserer sagen unødigt og er heldigvis overflødig, da følgende sætning kan vises: Dersom alle  $a_i$  og  $b_j$  er heltallige, da vil enhver optimal løsning have den egenskab, at alle positive  $x_{ij}$  ligeledes er heltal.

Hvis restriktionerne omformes til:

$$\begin{array}{rcccc}
 x_{11} + \dots + x_{1n} & & & & \leq a_1 \\
 & x_{21} + \dots + x_{2n} & & & \leq a_2 \\
 & & \cdot & & \cdot \\
 & & \cdot & & \cdot \\
 & & \cdot & & \cdot \\
 & & & x_{m1} + \dots + x_{mn} & \leq a_m \\
 x_{11} + & x_{21} + & \dots + & x_{m1} & \geq b_1 \\
 x_{12} + & x_{22} + & \dots & + x_{m2} & \geq b_2 \\
 \cdot & & & & \cdot \\
 \cdot & & & & \cdot \\
 \cdot & & & & \cdot \\
 x_{1n} + & x_{2n} + & \dots & + x_{mn} & \geq b_n
 \end{array}$$

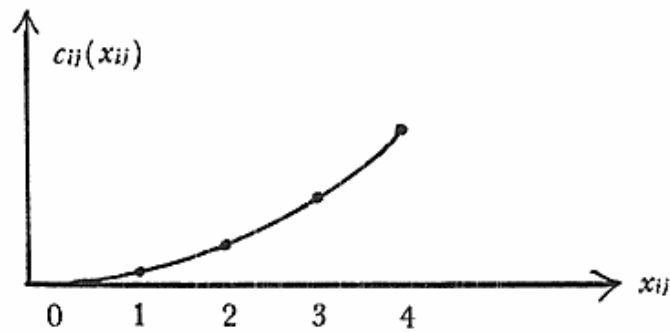
fremgår det klart, at vi står over for et sædvanligt lineært programmeringsprogram; blot vil man lægge mærke til, at netop to koefficienter i hver søjle i koefficientmatricen er  $\neq 0$ , og det er netop den omstændighed, der gør, at man ved løsningen forlader Simplex-metoden og i stedet søger andre veje.

Hvad løsningsmetoderne angår, skal vi i første omgang pege på de skræddersyede, hvoriblandt Ford & Fulkersons »Hungarian Method« (1962) skal fremhæves. Hvis enten antallet af produktionssteder eller forbrugere er lille, kan også Dynamisk Programmering anvendes, se f. eks. Pruzan (1965) eller Bellman & Dreyfus (1962), og endelig haves naturligvis Simplex-metoden, se f. eks. G. B. Dantzig (1963).

Det skal til slut bemærkes, at dersom der efterspørges mere, end der produceres, vil vi søge at tilfredsstille den del af efterspørgslen, der er lig den totale produktion, til minimal omkostning. Dette kan gøres ved at tilføje en ekstra fabrik, fra hvilken transport til ethvert marked er gratis, og hvis produktion er lig den overskydende efterspørgsel.

### 3. Klassisk transportproblem med konvekse omkostninger.

Hidtil er omkostningerne for enhver gren vokset proportionalt med den i grenen transporterede mængde. Hvis de proportionale omkostningskoefficienter  $c_{ij}$  erstattes af konvekse funktioner  $c_{ij}(x_{ij})$  og alt andet bevares, står vi med den variant, der er behandlet af E. M. L. Beale (1959).

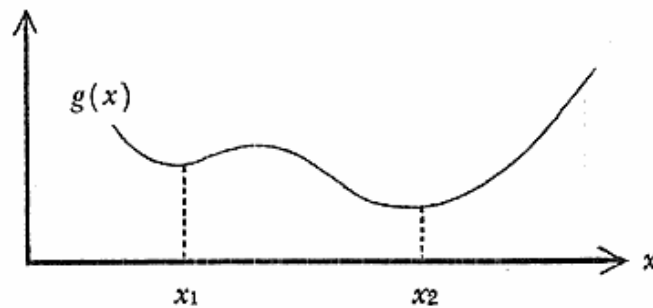


Konveks omkostningsfunktion:  
Ethvert punkt på en forbindelseslinie mellem to vilkårlige punkter på kurven ligger enten over eller på kurven.

#### 4. Fixed-cost Transportation Problem.

Det klassiske transportproblem samt dets første variant har begge haft det tilfælles, at de i en vis forstand »opførte sig pænt«. Hvad der ligger i denne vending har noget at gøre med begreberne *lokale* og *globale* optima.

I den klassiske analyse vil et maximum eller minimum for en funktion  $g(x)$  med en kontinuert differentialkvotient blive fundet i det punkt, hvor differentialkvotienten er nul. Dette kan måske resultere i punktet  $x = x_1$ , hvor  $g(x_1)$  er minimumsværdien for  $g(x)$  i et område omkring  $x_1$ ; dette



Funktion med to lokale og et globalt minimum.

kaldes et *lokalt minimum*. Af figuren ses imidlertid, at der findes et andet lokalt minimum for  $x = x_2$  samt at  $g(x_2) < g(x_1)$ . Vi kalder da  $g(x_2)$  for det *globale minimum*.

Hvis vi søger at minimere en konveks funktion, hvor de variable skal opfylde lineære uligheder, kan vi udnytte den omstændighed, at *ethvert lokalt minimum tillige vil være globalt*. Indtil nu er dette netop, hvad vi har gjort, men objektfunktionen i fixed-cost transportproblemet ser således ud:

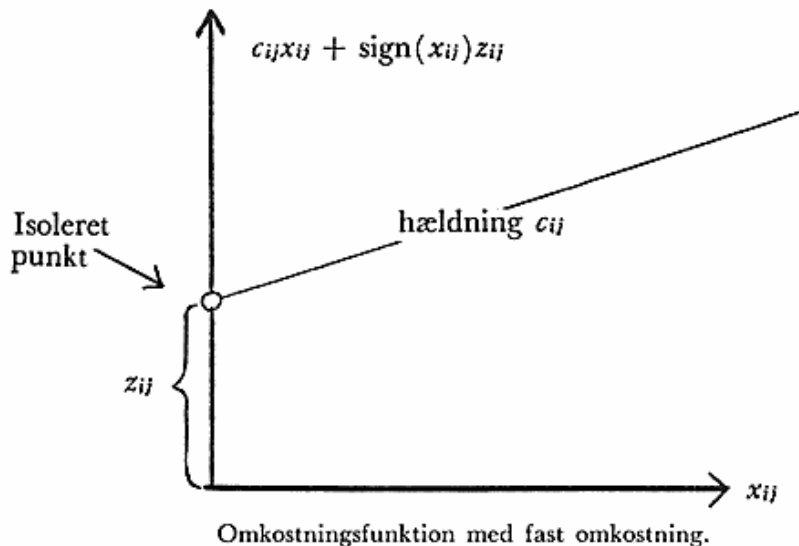
$$\min \sum_{i=1}^m \sum_{j=1}^n (c_{ij}x_{ij} + \text{sign}(x_{ij})z_{ij})$$

$$\text{hvor } \text{sign}(x_{ij}) = \begin{cases} +1 & \text{for } x_{ij} > 0 \\ 0 & \text{for } x_{ij} = 0 \\ -1 & \text{for } x_{ij} < 0 \end{cases}$$

og hvor  $z_{ij}$  er givne, ikke-negative tal.

Restriktionerne er uændrede fra afsnit 2.

Konstanterne  $z_{ij}$  kaldes *faste omkostninger* (fixed-costs, fixed charges), da objektfunktionen øges med et fast tillæg,  $z_{ij}$ , hvis  $x_{ij}$  er positiv, og dette tillæg er i så fald uafhængigt af  $x_{ij}$ . Den karakteristiske sammenhæng ses af figuren:



Fixed-cost problemer i almindelighed er såvel yderst realistiske som yderst ubehagelige at løse, og tilstedeværelsen af lokale optima *forskellige* fra det globale er netop den omstændighed, der bevirker at fixed-cost problemer kun i specielle tilfælde lader sig løse med traditionelle metoder.

Vi vil ikke her redegøre nærmere for arten af denne problematik, men blot henvise til diskussionen hos J. Krarup (1967). Af metoder til eksakt løsning af problemet findes så vidt vides kun den af J. Krarup udviklede (Branch and Bound), medens approximative metoder er angivet af bl. a. M. Balinski (1961). Mindre problemer kan dog også løses eksakt ved hjælp af dynamisk programmering som vist af Pruzan (1965).

### 5. Plant Location.

Lad der være givet  $m$  mulige placeringer for produktionssteder (fabrikker) samt  $n$  markeder, der alle skal forsynes med en og samme varetype. Efterspørgslen ved marked  $j$  betegnes med  $b_j$ , og  $k_i$  er den faste omkostning for etablering af fabrik  $i$ . Fra enhver potentiel fabrik findes direkte transportveje til ethvert marked, og  $h_{ij}(x_{ij})$  er transportomkostningerne som funktion af den transporterede mængde fra fabrik nr.  $i$  til marked nr.  $j$ . Specielt kan transportomkostningerne være lineære, d. v. s.  $h_{ij}(x_{ij}) = c_{ij}x_{ij}$ , hvor  $c_{ij}$  er prisen pr. enhed. Der er ikke foreskrevet nogen maximal produktionskapacitet for de enkelte fabrikker; enhver fabrik er principielt i stand til at forsyne samtlige markeder.

Problem: Bestem hvilke fabrikker, der skal etableres (åbnes), hvilke markeder de skal forsyne samt leverancernes størrelse, således at summen af de hermed forbundne faste omkostninger plus de variable transportomkostninger minimeres. Restriktioner: Den givne efterspørgsel skal imødekommes.

Løsningen angives i form af en  $m \times n$  matrix,  $x = \{x_{ij}\}$ , hvor  $x_{ij}$  er antallet af vareenheder (flow) fra fabrik nr.  $i$  til marked nr.  $j$ .

Vi betragter det lineære tilfælde:

Minimaliser  $g(x)$

$$= \sum_{i=1}^m \left( \sum_{j=1}^n c_{ij}x_{ij} + \text{sign} \left( \sum_{j=1}^n x_{ij} \right) k_i \right)$$

$$\sum_{i=1}^m x_{ij} = b_j; \quad j = 1, \dots, n; \quad x_{ij} \geq 0;$$

hvor  $b_j$  er positive heltal, og hvor  $c_{ij}$  og  $k_i$  er ikke-negative.

Det kan meget vel tænkes, at produktionsomkostningerne pr. vareenhed varierer fra fabrik til fabrik, når enten produktionsmetoderne kan være forskellige, eller når særlige geografiske forhold (vandkraft, variation i elpriser, toldbestemmelser etc.) gør sig gældende. Dette kan let inkluderes i modellen (idet produktionsomkostningerne blot opfattes som et tillæg til transportpriserne).

Enhver fabrik kan enten være lukket eller åben, dog må ikke alle fabrikker være lukkede samtidigt. Antallet af kombinationer af åbne og lukkede fabrikker må derfor være  $2^m - 1$ . Uanset hvilke fabrikker vi beslutter os til at åbne, kan transportomkostningerne herefter bestemmes direkte, idet ethvert marked da vil få sin totale forsyning fra den af de åbne fabrik-



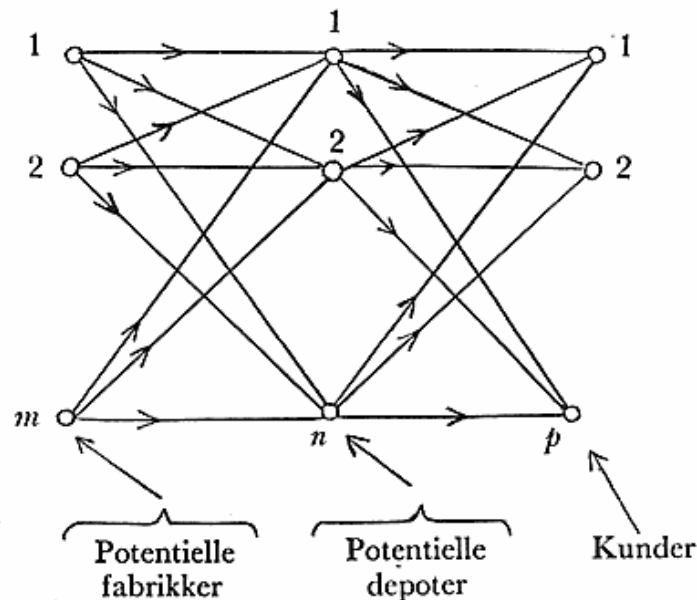
ker, hvor enhedsprisen er mindst. Dette indebærer, at det reelle antal mulige løsninger også er  $2^m - 1$ .

Det her formulerede problem er tydeligvis ikke-lineært på grund af de faste omkostningers tilstedeværelse. Naturligvis råder man idag også over adskillige metoder til løsning af ikke-lineære problemer; en anvendelse af sådanne metoder – dynamisk programmering indbefattet – forudsætter dog, at det forelagte problem besidder visse egenskaber. Det vil føre for vidt her at redegøre for den dybere liggende teoretiske baggrund, vi må blot postulere, at ingen af disse ikke nærmere specificerede egenskaber er til stede her.

Ved løsning på datamaskine synes den af Efrøymson & Ray (1966) angivne at være alle andre klart overlegen. Det anføres, at problemer med 50 fabrikker og 250 kunder i middel tager ca. 10 min. på IBM 7094. Efrøymson & Ray viser tillige, hvorledes den faste omkostning hørende til ethvert produktionssted kan erstattes af en vilkårlig konkav funktion. En manuel metode er udviklet af Bilde & Krarup (1967).

#### 6. »Location« med mellemtrin.

Dette problem er en generalisation af det i afsnit 5 omtalte (plant location). Nettet er her sammensat af to bipartitte dele som vist:



Mellem fabrikkerne og kunderne, hvis efterspørgsel er kendt, kan indskydes et antal depoter.

Omkostningerne for både fabrikker og depoter kan være enten faste eller blot en konkav funktion i almindelighed. For samtlige grene (trans-

portveje) er omkostningerne lineære. Så vidt vides har kun Branch and Bound teknikken kunnet opvise resultater på en rimelig regnetid. Der kan beklageligt nok ikke henvises til publikationer herom.

### 7. Transshipment.

Vi forlader nu det bipartitte net,  $K_{m,n}$ , der kun bestod af kilder og terminaler og går over til at betragte vilkårlige, orienterede net, hvor flow i visse mellemliggende punkter hverken tilføres eller forlader nettet, men blot videresendes (transshipment).

Nettet betegnes med symbolet  $(N; U)$ , hvor  $N$  er mængden af punkter og  $U$  er mængden af grene, og flow i gren  $(i,j) \in U$ ,  $i \in N$ ,  $j \in N$  kaldes  $x_{ij}$ .

Lad  $Y$  og  $Z$  være to delmængder af mængden  $N$ . Med symbolet  $(Y,Z)$  betegnes mængden af alle de grene i nettet, hvis startpunkt tilhører  $Y$ , og hvis slutpunkt tilhører  $Z$ .

Desuden indføres symbolet:

$$x(Y,Z) = \sum_{i \in Y, j \in Z, (i,j) \in U} x_{ij}$$

Under anvendelse heraf vil  $x(y,N)$  være den mængde flow, der udgår fra punkt  $y$  og  $x(N,y)$  er flowmængden, der løber til  $y$ .

I transshipmentproblemet har alle grene uendelig stor kapacitet, og til enhver gren  $(i,j) \in U$  er knyttet en given omkostningskoefficient,  $c_{ij}$ . Bestem en flowfordeling i nettet, der imødekommer en vis efterspørgsel i nogle punkter (terminalerne) fra forsyninger i andre (kilderne), som opfylder balancebetingelserne for de mellemliggende punkter og som minimerer de totale omkostninger.

Lad  $S$  og  $T$  være mængderne af hhv. kilder og terminaler, og lad  $R$  være mængden af de resterende, såkaldt mellemliggende punkter i nettet  $(N;U)$ .

$$\text{Minimaliser } \sum_U c_{ij}x_{ij}$$

under begrænsningerne

$$\begin{aligned} x(i,N) - x(N,i) &\leq a_i; & i \in S & \text{ Forsyningskapaciteten må ikke overskr.} \\ x(i,N) - x(N,i) &= 0; & i \in R & \text{ Flowbalance} \\ -x(j,N) + x(N,j) &\geq b_j; & j \in T & \text{ Efterspørgsel skal imødekommes} \end{aligned}$$

$$x_{ij} \geq 0, \text{ alle } (i,j) \in U$$

Anbefalelsesværdig løsningsmetode: En »skrabet« version af Ford & Fulkersons algoritme, »Out of Kilter«, der omtales i næste afsnit.

### 8. *Det generelle minimalomkostningsproblem.*

Hvis et net er både kilde- og terminalfrit, taler vi om *cirkulation* af flow. Til hver gren i det vilkårlige orienterede net  $(N;U)$  knyttes en omkostningskoefficient,  $c_{ij}$ , samt en nedre og øvre grænse,  $l_{ij}$  og  $u_{ij}$  for den mængde flow, der kan flyde i gren  $(i,j)$ .

Problem: Bestem en *cirkulation*, der minimerer

$$\sum_U c_{ij}x_{ij}$$

under begrænsningerne

$$l_{ij} \leq x_{ij} \leq u_{ij} \quad (i,j) \in U$$

samt  $x(i,N) - x(N,i) = 0$   $i \in N$   
(cirkulationsbetingelse)

Ved en mulig cirkulation i nettet forstås blot en cirkulation, der tilfredsstiller begrænsningerne. Betingelserne for eksistensen af en mulig cirkulation er udledt af A. J. Hoffman (1960).

Hvis alle data er heltallige, og hvis en brugbar cirkulation eksisterer, da kan en optimal cirkulation bestemmes med Ford-Fulkersons algoritme: Out-of-kilter. (Out-of-kilter kan oversættes ved: Ude af balance som modsætning til in-kilter, der er udtryk for en ligevægtstilstand). Algoritmen er første gang beskrevet i en artikel fra 1961 og senere gjort til genstand for en udvidet behandling i Ford & Fulkersons idag allerede klassiske værk fra 1962 om network flow.

Algoritmens meget generelle sigte fortjener at blive nævnt. Alle tidligere beskrevne problemstillinger samt et par af de efterfølgende kan uden vanskelighed transformeres til og løses som minimalomkostningsproblemer på cirkulationsform, uden at det dog hermed er sagt, at en sådan fremgangsmåde altid vil være den mest hensigtsmæssige i de pågældende tilfælde. En detaljeret redegørelse herfor kan findes hos Krarup (1966).

### 9. *Korteste vej.*

Ved en *vej* fra punktet  $s$  til punktet  $t$  i et givet, orienteret net forstås en følge af grene  $(s,y_1), (y_1,y_2), \dots, (y_k,t)$ . En vej, hvori alle punkterne  $s, y_1, y_2, \dots, y_k, t$  er forskellige, kaldes elementær. Dersom  $s$  og  $t$  falder sammen, kaldes vejen en *løkke*.

Det antages, at *længden* af enhver gren er kendt; »længde« kan i praksis

stå for såvel afstand som tid eller andet. Ved længden af en vej forstås blot summen af længderne af de grene, hvoraf veje  $n$  er sammensat. Der pålægges ikke de enkelte grenes længder andre restriktioner end den, at længden af enhver løkke skal være ikke-negativ. Idet  $s$  og  $t$  er to givne punkter, vil vi stille fem spørgsmål:

Bestem:

- a) Den korteste vej fra  $s$  til  $t$ .
- b) Den  $k$ 'te korteste vej fra  $s$  til  $t$ .
- c) De korteste veje fra  $s$  til ethvert andet punkt.
- d) De korteste veje mellem ethvert punktpar i nettet.
- e) De korteste veje fra ethvert punkt i en delmængde af punkter til ethvert punkt i en anden delmængde.

Der findes talrige algoritmer til bestemmelse af korteste veje. Hvad a), b) og c) angår, synes dynamisk programmering at være det mest virkningsfulde værktøj, og de simple »network flow« metoder, der er angivet af Ford & Fulkerson p. 131 ligner også dynamisk programmering til forveksling. For b)'s vedkommende kan der yderligere henvises til Bellman og Kalaba.

Tilfældene d) og e) kan let medføre store lagerkrav ved løsning på datamaskine. d) er behandlet af Farbey, Land & Murchland (1967), hvis »Cascade Algorithm« senere er revideret af Bilde & Krarup (1967).

Endelig følger c), som bl. a. har stor betydning ved bestemmelse af trafikbelastning på et givet vejnet. Lagerkravene kan ved en særlig teknik angivet af Land & Stairs (1967) holdes nede på et rimeligt niveau, idet »uinteressante« korteste veje ikke beregnes.

#### 10. *Travelling Salesman.*

I al sin enkelhed kan dette berømte eller måske snarere berygtede problem formuleres sålunde: En handelsrejsende ønsker med udgangspunkt i sin hjemby at besøge  $n-1$  andre byer for derefter at returnere til sit udgangspunkt. Afstanden fra enhver by til de  $n-1$  øvrige byer er givet, og ruten ønskes bestemt således, at den totale rejseafstand minimeres.

Idet byerne nummereres  $1, 2, \dots, n$ , vil vi med  $c_{ij}$  betegne afstanden fra by nr.  $i$  til by nr.  $j$ . For alle  $i$  sættes  $c_{ij} = \infty$ . Endvidere sættes  $x_{ij} = 1$ , såfremt den handelsrejsende fra by nr.  $i$  fortsætter til by nr.  $j$ ; i modsat fald sættes  $x_{ij} = 0$ . Den handelsrejsendes problem kan herefter udtrykkes ved følgende program:

Bestem de  $n \times n$  ubekendte,  $x_{ij}$   
der minimerer

$$\sum_{i=1}^n \sum_{j=1}^n c_{ij}x_{ij} \quad (1)$$

under begrænsningerne

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

$$x_{ij} = 0 \text{ eller } 1 \quad (3)$$

samt

$$x_{i1,i2} + x_{i2,i3} + \dots + x_{ik,i1} \leq k-1 \quad (4)$$

hvor  $(i1, i2, \dots, ik)$  er en vilkårlig permutation af tallene  $1, 2, \dots, k$ ,  $2 \leq k < n$ .

Betingelse (2) sikrer, at der til enhver by findes netop 1 indgang og 1 udgang, og betingelse (4) sikrer mod en eventuel returnering til en allerede besøgt by, førend alle øvrige byer er passerede.

Afstandene kan bekvemt beskrives som elementerne i en  $(n \times n)$  matrix  $c = \{c_{ij}\}$ . Ved en mulig løsning (en mulig rute) forstås en  $(n \times n)$  matrix  $x = \{x_{ij}\}$ , hvis elementer tilfredsstiller begrænsningerne (2)–(4). Ved en optimal løsning (en optimal rute) forstås en mulig løsning, der tillige minimerer (1).

Det indses umiddelbart, at antallet af mulige ruter i det generelle tilfælde må være  $(n-1)!$ ; fra et udgangspunkt, der kan fastsættes vilkårligt, kan den første by vælges på  $n-1$  måder, den næste kan derefter vælges på  $n-2$  måder og så fremdeles. I det symmetriske tilfælde, der er karakteriseret ved  $c_{ij} = c_{ji}$  halveres antallet, idet enhver mulig rute kan gennemløbes i to retninger.

Eksistensen af en optimal, men ikke nødvendigvis centrydig rute er ubestridelig. Det er ligeledes evident, at en optimal rute altid kan findes – såfremt man systematisk afprøver samtlige muligheder. Men betragter man blot et symmetrisk tilfælde af forholdsvis beskeden størrelse med f. eks. 16 byer, vil antallet af mulige ruter blive 653 837 184 000, så det egentlige problem består i at udvikle en algoritme, der på en effektiv måde forkaster et stort antal muligheder og blandt de resterende få bestemmer en optimal.

Adskillige problemer af vidt forskellig natur munder ud i en bestemmelse af en optimal permutation af  $n$  forskellige tal blandt  $n!$  mulige permutationer eller som en variant heraf. Typiske problemstillinger af denne art kendes fra produktionsplanlægning, hvor f. eks. opstillingstiden for en given maskine før behandlingen af et givet emne afhænger af det senest

bearbejdede emne ved denne maskine; fra planlægning og drift af distribueringsystemer; fra design-situationer, hvor  $n$  objekter søges sammenkoblet på en måde, hvorved den sammensatte enhed opfylder visse egenskaber (de fleste kan vel endnu genkalde sig desperationen under samling af puslefanter). Det er derfor forståeligt, at problemet bortset fra den rent akademiske interesse har fascineret mange, som så har investeret både tid og arbejdskraft i forsøg på opnåelse af en løsning. En overgang udartede det til næsten national feber i U. S. A., da et sæbefirma som led i en reklamekampagne udsatte præmier på indtil 10.000 \$ for bestemmelse af en optimal rute mellem 33 byer af kendt beliggenhed.

Vi skal ikke her i detaljer drøfte den mangfoldighed af forslag, der er fremkommet i årenes løb siden 1930, hvor grafteoretikeren Karl Menger ved et seminar i Wien introducerede problemet for første gang; en oversigt over de forgæves anstrengelser kan findes i en artikel af M. M. Flood (1956), der gør status op og bl. a. konstaterer det helt indlysende faktum, at en optimal rute aldrig kan skæres sig selv.

En algoritme, der baseret på dynamisk programmering løser problemer for  $n \leq 10$ , er udviklet af Gonzales (1962). Køretiden vokser eksponentielt med antallet af byer, og der rapporteres, at  $n = 5$  tager ca. 10 sekunder,  $n = 10$  tager omkring 8 minutter (IBM 1620) samt at tilføjelse af blot een by bevirker en multiplikation af køretiden med en faktor, der for  $n = 10$  er ca. 3. Lagerkravene vokser med tilsvarende hast. Også Held & Karp (1962) har med dynamisk programmering løst problemer med op til 13 byer på IBM 7090. 13 byer tager 17 sekunder, og med en eksponentiel vækst som hos Gonzales vil 20 byer kræve ca. 10 timers regnetid, hvilket dog ikke har været forsøgt på grund af de enorme lagerkrav. For  $n > 13$  anviser Held & Karp en approximativ procedure, der synes at fungere ganske godt, men som naturligvis ikke garanterer et globalt optimum.

Det største fremstød i de senere år skyldes J. Little et al., der i 1963 offentliggjorde en algoritme, som løste problemet ved rendyrket anvendelse af en teknik, som første gang er skitseret af Land & Doig (1960), og som Little selv har navngivet: Branch and Bound.

Algoritmens kvalitet fremgår af følgende skema:

Antal byer	Antal løste problemer	Køretid i minutter (IBM 7090)	
		Middeltal	Spredning
10	100	0.012	0.007
20	100	0.084	0.063
30	100	0.975	1.240
40	5	8.37	10.2

hvoraf det ses, at køretiden omtrentlig vokser med en faktor 10, når antallet af byer øges med 10.

### 11. *Truck Dispatching Problem.*

Travelling Salesman problemet kan generaliseres på forskellige måder ved tilføjelse af passende tillægsbetingelser. Man kan opnå en variant ved at kræve, at den handelsrejsende returnerer til sit udgangspunkt, hver gang han har besøgt  $m$  byer, hvor  $m$  er divisor af  $(n-1)$ . For givne værdier af  $m$  og  $n$  består problemet i at bestemme  $(n-1)/m$  løkker, der alle har et givet punkt fælles, således at summen af samtlige løkkers længde er minimum; denne generalisation omtales undertiden som »Clover Leaf Problem«. Hvis  $m$  er lille, kan man undertiden bestemme optimale grupper af  $m$  sammenhørende punkter ved direkte studier af et landkort, hvor de relevante vejstrækninger er vist. Man opsøger i så fald »punktklynger« og kan ved »trial and error« metoder finde en næsten-optimal løsning, idet der tages hensyn til, at ingen løkker skærer sig selv. Hvis  $m$  derimod er stor, eller hvis det tilstrækkelige antal klynger ikke findes, er en sådan fremgangsmåde normalt ikke at anbefale.

En anden generalisation fører os til Truck Dispatching problemet, der i den originale præsentation handler om et benzinselskabs strategi under bestemmelse af optimale ruter for et antal tankvogne, der forsyner en række servicestationer fra et centraldepot.

Givet:

- a)  $n$  servicestationer,  $P_1, P_2, \dots, P_n$  samt et depot,  $P_0$ .
- b) En afstandsmatrix  $\{c_{ij}\}$ , hvor  $c_{ij}$  er afstanden fra  $P_i$  til  $P_j$ ;  
 $i, j = 0, 1, \dots, n$ .
- c)  $b_i$ : Efterspørgsel ved stationen  $P_i$ ,  $i = 1, \dots, n$ .
- d)  $Q$ : Kapacitet for tankvogne (alle tankvogne forudsættes at have samme kapacitet).

Det antages, at  $Q > \max_i \{b_i\}$ .

En approximativ procedure er udviklet af Dantzig og Ramser (1958); denne metode kan også anvendes til det ovenfor omtalte »kløverbladsproblem«. Blandt nyere arbejder skal nævnes P. Bertier, der i 1966 beskriver en Branch and Bound algoritme til exakt løsning.

### 12. *Multi-commodity flow.*

Et karakteristisk træk ved alle hidtil betragtede problemer har været, at kun en enkelt »varetype« flyder i det givne net (single-commodity). Der findes i rigt mål litteratur herom, hvorimod litteraturen er særdeles forbe-

holden, når to eller flere varetyper flyder simultant (multi-commodity). Vi kan forestille os et generaliseret transportproblem, hvor hver fabrik producerer flere varetyper, og hvor kundernes efterspørgsel ikke blot karakteriseres ved den totale mængde men antal enheder af hver varetype.

Der er dog flere måder, hvorpå disse vanskeligheder kan omgås. Det er således ofte muligt at transformere problemet til et single-commodity problem og derefter anvende en af de tidligere nævnte metoder. Desuden kan dynamisk programmering hyppigt benyttes; se f. eks. næste afsnit, og endelig kan man – hvis omkostningerne ellers er lineære – vende tilbage til Simplex-metoden.

Drejer det sig kun om opnåelse af en approximativ løsning, vil en passende kombination af matematisk programmering og simulation ofte være hensigtsmæssig.

### 13. *The Many-Product Cargo Loading Problem.*

Dette sidste problem i rækken drejer sig om et transportfirmas optimale udnyttelse af de til rådighed værende køretøjsmidler, og problemet medtages her som modvægt til de lidt nedslående bemærkninger i forrige afsnit. Såvel formulering og løsning skyldes Pruzan & Jackson (1967), og problemet kan opfattes som en ret kompliceret kombination af multi-commodity flow og et specielt transportproblem med faste omkostninger. I det aktuelle tilfælde haves over 500 forskellige varetyper og mere end 40 lokaliteter, hvor losning og lastning kan finde sted; løsningsmetoden er baseret på dynamisk programmering.

### 14. *Konklusion.*

Som et passende sammendrag af de forrige afsnit vil vi her til slut vise et skema med oversigt over forskellige løsningsmetoders »effekt«, idet tallet »1« står for den mest anbefalelsesværdige teknik i det pågældende tilfælde. De tomme pladser antyder direkte uegnede kombinationer.

Stort set er kun de eksakte løsningsmetoder fremhævet, men der er naturligvis intet til hinder for anvendelse af approximative metoder i de situationer, hvor eksakte metoder ved løsning af et problem på datamaskine er for langsomme og/eller stiller urimelige lagerkrav. En sådan fremgangsmåde er sandsynlig for problemer, ud for hvilke der er markeret et kryds i sidste kolonne i skemaet.

Det skal sluttelig understreges, at de i disse noter beskrevne problemstillinger på ingen måde udgør en komplet samling; vi har blot søgt at foretage et skønsomt udvalg blandt nogle af de fra teori og praksis mest kendte.



	EXAKTE METODER						APPROXIMATIVE METODER
	Lineær programmering (Simplex)	Dynamisk programmering	All-integer (mixed-integer) programmering	Branch and Bound	Specielle algoritmer	Diverse network flow algoritmer	
Klassisk transportproblem	2	3				1	
- med kapacitetsbegrænsning	2	3				1	
- med konvekse omkostn.		2			1		
- med faste omkostninger		2	3	1			x
Plant Location		2	3	1			x
»Location« med mellemtrin		1-2		1-2			x
Transshipment	2					1	
Generelt min.omk. problem	2					1	
Korteste vej a) b) c)		1-2				1-2	
- - d) e)		2			1		
Travelling Salesman		2		1			x
Truck Dispatching		2		1			x
Multi-commodity flow	1-2	1-2				3	x
Many-Prod. Cargo Loading		1					x

»Effektivitetsmatrix«

*Referencer:*

- M. L. Balinski:  
Fixed-cost Transportation Problems. Nav. Res. Log. Quart. Vol. 8, 1961.
- E. M. L. Beale:  
An Algorithm for solving the Transportation Problem when the Shipping Cost over each Route is Convex. Nav. Res. Log. Quart. Vol. 6, 1959.
- R. Bellman & S. Dreyfus:  
Applied Dynamic Programming. Princeton University Press, 1962.
- R. Bellman & R. Kalaba:  
On  $k$ 'th best Policies. Journal of the SIAM, Vo. 8, 1960.
- P. Bertier:  
Procédures pour élaborer des tournées de distribution. METRA, Vol. 8, 1966.
- O. Bilde & J. Krarup:  
En manuel metode til optimering af produktionssteders beliggenhed. Institutet for Matematisk Statistik og Operationsanalyse, DtH, 1967.
- O. Bilde & J. Krarup:  
A Modified Cascade Algorithm for Shortest Paths. Institutet for Matematisk Statistik og Operationsanalyse, DtH, 1967.
- G. B. Dantzig:  
Linear Programming and Extensions. Princeton University Press, 1963.
- G. B. Dantzig & J. H. Ramser:  
The Truck Dispatching Problem. Operations Research, Vol. 6, 1958.
- M. A. Efraymson & T. L. Ray:  
A Branch-Bound Algorithm for Plant Location. Operations Research, Vol. 14, 1966.
- B. A. Farbey, A. H. Land & J. D. Murchland:  
A Cascade Algorithm for Shortest Paths. Management Science, Vol. 14, No. 1, 1967.
- M. M. Flood:  
The Traveling Salesman Problem. Operations Research, Vol. 4, 1956.
- L. R. Ford & D. R. Fulkerson:  
Flows in Networks. Princeton University Press, 1962.
- R. H. Gonzales:  
Solution of the Traveling Salesman Problem by Dynamic Programming on the Hypercube. Interim Technical Report No. 18, M. I. T., 1962.

- M. Held & R. M. Karp:  
A Dynamic Programming Approach to Sequencing Problems. *Journal of the SIAM*, Vol. 10, 1962.
- A. J. Hoffman:  
Some Recent Applications of the Theory of Linear Inequalities to Extremal Combinatorial Analysis. *Proc. Symp. on Appl. Math.*, 1960.
- J. Krarup:  
Fixed-Cost Network Flow Problems. *Instituttet for Matematisk Statistik og Operationsanalyse, DTH*, 1967.
- J. Krarup:  
Om løsning af lineære cirkulationsproblemer. *Erhvervsøkonomisk Tidsskrift*, nr. 3, 1966.
- A. H. Land & A. G. Doig:  
An Automatic Method of Solving Discrete Programming Problems. *Econometrica*, Vol. 28, 1960.
- A. H. Land & S. W. Stairs:  
The Extension of the Cascade Algorithm to Large Graphs. *Management Science*, Vol. 14, No. 1, 1967.
- J. D. C. Little et al.:  
An Algorithm for the Traveling Salesman Problem. *Operations Research*, Vol. 11, 1963.
- P. M. Pruzan:  
Om anvendelse af dynamisk programmering. *Ingeniøren*, nr. 17, 1965.
- P. M. Pruzan & J. T. R. Jackson:  
The Many-Product Cargo Loading Problem. *Nav. Res. Log. Quart.*, Vol. 14, 1967.