Simulering av linjefart[1)]

AV

VITENSKAPLIG ASSISTENT FINN KYDLAND,

NORGES HANDELSHØYSKOLE

I. Innledning.

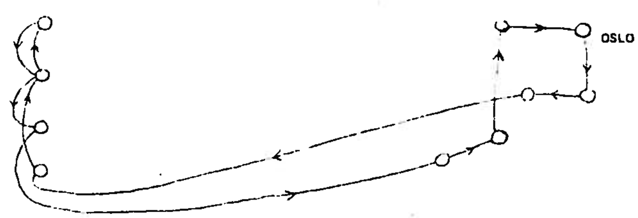
Linjefart er karakterisert ved at et bestemt antall skip går i fast rute på et bestemt antall havner i en bestemt rekkefølge, f. eks. mellom havner i Europa og havner på vestkysten av Nord-Amerika. Skipene frakter som oftest forskjellige varer i begrensede mengder som befrakterne ikke vil chartre eget skip for. Rederens problem blir å velge havner hvor lasttilstrømningen er tilstrekkelig stor og jevn over tiden slik at havnene passer inn i en fast rute, og å fastlegge den beste rekkefølge mellom disse. Driftsresultatet vil kunne avhenge av handlingsparametre som antall skip, deres karakteristika og den ruteplan de går etter.

Lasttilstrømningen i hver havn vil kunne variere meget over tiden (f. eks. sesongmessig), og det er vanskelig å tilpasse en ruteplan som både klarer å ta det meste av toppene og har noenlunde kapasitetsutnyttelse i tider med lavere tilstrømning. En mulig løsning er å chartre et ekstra skip fra en annen reder i toppsesongen som innpasses i ruteplanen. Man har likevel problemet å bestemme hvor mange skip som skal gå fast i ruten.

Hvis man kan skaffe data over lasttilstrømningen i hver havn, f. eks. ved at et rederi allerede har hatt skip gående noen tid på de aktuelle havner, eller ved at man på andre måter kan komme frem til noenlunde pålitelige prognoser for lasttilstrømningen, skulle der være muligheter for besparelser og/eller øket inntjening hvis man på en eller annen måte kunne optimalisere sine handlingsparametre. Simulering peker seg ut som en brukbar metode for analyse av dette optimaliseringsproblem, fordi man kan ta hensyn til flere usikkerhetsfaktorer, f. eks. lasttilstrømning, ventetider eller dødtider i havnene, fraktrater osv.[2]

II. Oversikt over modellen.

Ved utarbeidelsen av simuleringsmodellen har vi tatt utgangspunkt i et konkret ruteopplegg som kan være representativt for flere andre ruter. Denne rute omfattet et antall havner i Europa og et tilsvarende antall på vestkysten av Nord-Amerika. Dette kan illustreres slik:



Vi har altså havnene samlet i to geografiske områder, hvor avstanden mellom de to områdene er stor i forhold til avstandene mellom havner innen områdene. På figuren er tegnet inn et eksempel på besøksrekkefølge. Valg av besøksrekkefølge vil åpenbart kunne være avgjørende for resultatet av rutens drift. Ser man på havnene på vestkysten av Nord-Amerika, kan båtene f. eks. gå innom hver havn kun en gang, enten på borttur eller på hjemtur. Men det kan også tenkes at det vil svare seg å anløpe enkelte havner på borttur for å losse, og så gå innom igjen på hjemtur og laste, for derved å

[1] I denne artikkel er det meningen å beskrive en simuleringsmodell for linjefart som er utviklet etter initiativ fra Albert Harloff ved A.s Bergens Mekaniske Verksteder. Støtte til dette prosjekt er blitt gitt av A.s Bergens Mekaniske Verksteder, Norges Handelshøyskole og Norges Naturvitenskapelige Forskningsråd.

[2] En foreløpig beskrivelse finnes i Doksrød *et al.*, «En simuleringsmodell for linjefart», NHH Bergen, 1967.

mellom de ekstra inntekter man kan få ved øket lastmengde og de ekstra kostnader, f. eks. faste havneutgifter, som påløper ved å anløpe havnen to ganger istedenfor en. I simuleringsmodellen kan man undersøke dette ved å prøve alternative besøksrekkefølger som man selv spesifiserer, og i tilfelle dobbeltanløp sørger modellen automatisk for at varer losses i vedkommende havn på borttur og lastes på tilbakeveien.

Lasttilstrømningen i havnene vil bestemme hvilket antall båter det er optimalt å bruke og dermed også anløpsfrekvensen. Varetypene kan være avgjørende for om det vil svare seg å bruke skip med spesiallasterom, f. eks. beregnet på også å kunne ta kjølelast. Her er det åpenbart en rekke faktorer som påvirkes gjensidig. Det optimale antall båter kan være forskjellig for forskjellige besøksrekkefølger, og det optimale antall vil samtidig kunne variere med skipstype. Der må på en eller annen måte skje en simultan tilpasning av besøksrekkefølge, av antall skip og deres karakteristika, som hastighet, dødvekt, lastvolum, volum for spesiallasterom og laste- og losseutstyr. Man må altså nøye seg med en eller annen form for suboptimalisering. Dette gjøres i modellen på følgende måte: Man tar utgangspunkt i en bestemt skipstype. Man velger så på forhånd de besøksrekkefølger som ser brukbare ut og som man vil vurdere mot hverandre. Innenfor hver besøksrekkefølge gjøres så mange simuleringer man ønsker med forskjellige antall skip (en simulering tilsvarer en rundtur med hvert skip), og for hver besøksrekkefølge registreres det optimale antall skip. Til slutt står man igjen med en optimal besøksrekkefølge og et optimalt antall skip for denne.

Siden kan man forandre skipsdata og gjøre samme operasjon igjen. Dette kan gjøres for så mange skipstyper man ønsker, og på denne måten vil man også få et inntrykk av hvilken skipstype som er optimal.

Kriteriet som brukes ved optimaliseringen, er nettofortjeneste pr. døgn med tidshorisont lik tiden for en rundtur. Rundturtiden blir også et resultat av optimaliseringen. Den varierer med varetyper og -mengder, fordi forskjellige varer trenger forskjellig laste- og lossetid pr. tonn. Det som gjøres, er altså å maksimere nettofortjeneste pr. døgn for hver rundtur under ett. For å kunne gjøre dette, må vi ha en utgangshavn hvor båtene alltid er tomme før de begynner å laste igjen. Tiden for en rundtur regnes da fra båten begynner å laste i utgangshavnen til neste gang den begynner å laste i samme havn etter å ha anløpt de andre havnene i ruten (i vårt eksempel brukes Oslo som utgangshavn; se figur 1). Dette kan kanskje være urealistisk for enkelte ruter. Hvis man ønsker å lempe på dette krav, vil vi vanskelig kunne få noen eksakt av tilnærming kan være stor hvis man som utgangshavn velger den havn med gjennomsnittlig minst varer i transitt, og også tar hensyn til om varene som fraktes i transitt over denne havn gir så stort dekningsbidrag at det vanligvis vil være utvilsomt at de skal fraktes.

Under simuleringen blir varetilstrømningen generert tilfeldig på grunnlag av tall for forventning og spredning. Varegenereringen for hver simulering skjer for alle varer samtidig før båten begynner å laste i utgangshavnen. Før hver rundtur velger så modellen ut de lasttyper og -mengder som maksimerer nettofortjenesten pr. døgn.

Dødtider i havnene vil kunne variere en del, både avhengig av på hvilket tidspunkt på døgnet eller i uken båten ankommer og av andre forhold i havnene. Disse ventetidene kan man generere stokastisk for hver havn på grunnlag av data om forventning og spredning for dødtidene.

III. Modellens kjerne: optimaliseringsdelen.

De symboler som brukes i beskrivelsen er følgende:

Servicehastighet	s	(knop)
Utseilt distanse pr. rundtur	d	(mil)
Lastekapasitet	K	(tonn)
Volum totalt	V	(cf)
Andre volumbeskrankninger	$V_1, V_2, \ldots$	(cf)
Faste kostnader for skip	C_1	(kr./dag)
Tilleggskostnader på sjøen (bunkerskostn. etc.)	C_2	(kr./dag)
Faste havnekostnader i havn nr. i	H_i	(kr./anløp)
Lastekostnader for vare nr. j	k_j	(kr./tonn)
Lossekostnader for vare nr. j	k_j'	(kr./tonn)
Lastehastighet for vare nr. j	l_j	(tonn/døgn)
Lossehastighet for vare nr. j	l_j'	(tonn/døgn)
Stuingsfaktor for vare nr. j	q_j	(cf/tonn)
Fraktrate for vare nr. j	f_j	(kr./tonn)
Ventetid i havn nr. i (kan genereres stokastisk)	v_i	(døgn)
Antall varetyper	n	
Antall havner i besøksrekkefølgen	m	
Mengde av vare nr. j tilgjengelig (genereres stokastisk)	w_j	(tonn)
Fraktet mengde av vare nr. j	x_j	(tonn)

Med en vare menes her en varetype som fraktes fra en bestemt havn til en annen bestemt havn. Opplysninger om hvilke havner varene fraktes mellom, leses inn i modellen som egne data. Hvis en bestemt varetype fraktes fra, eventuelt til forskjellige havner, vil denne få en ny indeks for hvert skipningssted og/eller bestemmelsessted.

Med faste kostnader for skip mener vi summen av alle kostnader som ikke påvirkes av om skipet seiler eller ikke. Det kan være avskrivning, lønn til mannskap, forsyninger, forsikring (ikke for lasten), administrasjonskostnader etc. Disse forutsettes å kunne fordeles over tiden med et fast beløp pr. døgn.

Optimaliseringsdelen i modellen maksimerer nettofortjeneste pr. døgn for et skip for en rundtur under ett.

Ved bruk av ovenstående symboler, kan vi beregne følgende uttrykk:

Faste kostnader pr. rundtur uavhengig av last mengde (med negativt fortegn):

$$a_0 = -(C_1 + C_2) \cdot \frac{d}{s \, . \, 24} - \sum_{i=1}^{m} H_i - C_1 \cdot \sum_{i=1}^{m} v_i$$

Når vi her for letthets skyld skriver $\sum_{i=1}^{m}$, så er det å tolke som at i går fortløpende over alle havnene i den aktuelle besøksrekkefølge, og i tilfelle dobbeltanløp som vi tidligere har omtalt, vil en havn kunne svare til to i-er i ovenstående sumuttrykk.

Dekningsbidrag pr. tonn for vare nr. j:

$$a_j = f_j - k_j - k_j' - C_1\left(\frac{1}{l_j} + \frac{1}{l_j'}\right), \quad j = 1, \ldots\ldots, n.$$

Dekningsbidraget er her definert for vårt formål, idet vi også trekker fra de faste kostnader pr. døgn som påløper i det tidsrom som går med til å laste og losse et tonn av varen.

Tid i sjøen + ventetider (dvs. fast tidsforbruk pr. rundtur uavhengig av lastmengde):

$$\beta_0 = \frac{d}{s \, . \, 24} + \sum_{i=1}^{m} v_i$$

β_0 måles i døgn

Laste- + lossetid for vare nr. j:

$$\beta_j = \frac{1}{l_j} + \frac{1}{l_j'}, \quad j = 1, \ldots\ldots, n.$$

$\beta_j, j = 1, \ldots, n$, måles i døgn/tonn.

Vi ønsker nå å maksimere følgende uttrykk:

$$P = \frac{a_0 + a_1 x_1 + \ldots\ldots + a_n x_n}{\beta_0 + \beta_1 x_1 + \ldots\ldots + \beta_n x_n},$$

hvor a-ene og β-ene er de størrelser som er definert ovenfor, og hvor a_0 og β_0 er konstanter.

P kan maksimeres ved hjelp av en modifisert utgave av parametrisk lineær programmering.[3]

For å kunne stille opp objektfunksjonen på denne form, må vi gjøre noen forenklinger. Vi ser bort fra ikke-lineære faktorer i forbindelse med lasting og lossing. Da der bare arbeides på bestemte tider av døgnet, eventuelt på overtid mot ekstra betaling, kan laste- og lossetid og også laste- og lossekostnader variere alt etter på hvilket tidspunkt på døgnet båten kommer til havnen. For å gjøre problemet løsbart med ovennevnte metode, regner vi med et bestemt antall arbeidstimer pr. døgn i hver havn, dvs. at det kan lastes eller losses en viss mengde av hver varetype pr. døgn.

Blant de kostnader som regnes som faste uansett lastmengde, er bunkerskostnaden åpenbart ikke fast, men det er vanskelig å gjøre noe annet med den. Denne kostnad vil øke med økende last, dog ikke proporsjonalt. En del av denne kostnad skulle da egentlig trekkes fra hver vares dekningsbidrag varierende med varens vekt, hvor langt den skal fraktes, og hvor full båten er på forhånd. Det ville kunne påvirke valg av varer som skal fraktes. Forsøk viser imidlertid at de tallmessige forskjeller blir så små at det ytterst sjelden ville påvirke valg av varer, og da kun av de varer hvis fraktrate og andre karakteristika gjør at de ligger akkurat på grensen til å bli tatt med. Dette betyr i praksis at det er omtrent likegyldig hvilken av disse to, eventuelt flere varer, som blir tatt med.

En annen sak er at de totale kostnader, og dermed nettofortjenesten, kan avvike betydelig fra det riktige beløp dersom man går ut fra et fast tall for bunkerskostnaden pr. dag, dvs. for en fast lastvekt, istedenfor å beregne den særskilt mellom hvert par av havner på grunnlag av virkelig fraktet tonnmengde, spesielt dersom volumet blir den effektive begrensningen. Dette kan man ta hensyn til ved for hvert skip på forhånd å lese inn i modellen opplysninger om bunkerskostnader pr. døgn for alternative dødvekter, og så interpolere mellom disse for å finne de tilnærmet riktige kostnader.

Beskrankningene kan man betrakte som oppbygget av flere blokker.

1) Først har vi beskrankninger som sier at det som kan lastes av hver vare, må være mindre enn eller lik det som er tilgjengelig av vedkommende vare, dvs.

$$x_i \leq w_i, \qquad i = 1, \ldots\ldots, n,$$

hvor altså alle w_i, $i = 1, \ldots, n$, er generert stokastisk før hver rundtur.

2) Vektbeskrankninger for hver havn:

$$\begin{matrix}\text{Alle lastede} \\ \text{varer}\end{matrix} + \begin{matrix}\text{alle varer} \\ \text{i transitt}\end{matrix} - \begin{matrix}\text{alle lossede} \\ \text{varer}\end{matrix} \leq K$$

eller

$$\sum_{i=1}^{m} a_{n+i,j} x_j \leq K, \qquad i = 1, \ldots\ldots, m.$$

Her er $a_{n+i,j} = 1$ hvis vare nr. j kan fraktes fra havn nr. i til havn nr. $i+1$ (også i transitt), ellers 0.

[3] Se Appendiks 1.

3) Volumbeskrankninger for hver havn:

$$\frac{\text{Alle lastede}}{\text{varer}} + \frac{\text{alle varer}}{\text{i transitt}} - \frac{\text{alle lossede}}{\text{varer}} \leq V$$

eller:

$$\sum_{i=1}^{m} a_{n+m+i,j} x_j \leq V, \qquad i = 1, \ldots\ldots, m.$$

Her er $a_{n+m+i,j} = q_j$, dvs. lik vare nr. j's stuingsfaktor hvis vare nr. j kan fraktes fra havn nr. i til havn nr. $i - 1$, ellers 0.

Videre f. eks.:

$$\frac{\text{All lastet}}{\text{kjølelast}} + \frac{\text{all kjølelast}}{\text{i transitt}} - \frac{\text{all losset}}{\text{kjølelast}} \leq V_1$$

eller:

$$\sum_{i=1}^{m} a_{n+2m+i,j} x_j \leq V_1, \qquad i = 1, \ldots\ldots, m.$$

Her er $a_{n+2m+i,j} = a_{n+m+i,j}$ hvis vare nr. j er kjølelast, ellers 0.

Det som står ovenfor om frakting fra havn nr. i til havn nr. $i + 1$, må modifiseres for havn nr. m. Da erstattes $i + 1$ med 1, dvs. utgangshavnen.

IV. Duale evaluatorer.

Ved maksimering av et lineært programmeringsproblem får man samtidig som biprodukt et sett duale evaluatorer, en for hver beskrankning, som sier hvor meget objektfunksjonen (f. eks. fortjenesten) ville øke pr. enhet økning i vedkommende beskrankning.[5] Ved å summere de respektive duale evaluatorer kan man finne ut hva resultatet ville bli av en økning av flere beskrankninger samtidig. Dette er informasjon som anvendt på rette måte kan være meget nyttig.[6]

I vårt spesielle problem ville vi tilsvarende være interessert i hvor meget nettofortjenesten pr. døgn ville øke pr. tonn økning i dødvekt og/eller pr. cf økning i totalt volum eller kjølevolum. Den ikkelineære objektfunksjon gjør at de duale evaluatorer ikke kan bestemmes direkte fra det optimale simplextableau som i det lineære tilfelle, men de kan lett beregnes.[7]

Etter hver simulering kan man altså få en slags intern pris på innsatsfaktorene vekt og volum. Disse tallene kan fortelle hvor meget vi høyst ville være villig til å betale pr. døgn for ekstra enheter (tonn eller cf) av disse. Et annet eksempel på hva disse tallene kan brukes til, er sammenligning mellom forskjellige typer lastevolum. Hvis f. eks. den duale evaluator for kjølevolum er høyere enn for totalt volum, bør åpenbart kjølevolum få tildelt større andel av det totale volum.

De duale evaluatorer gir informasjon som man ellers bare ville fått ved alternative simuleringer hvor man testet forskjellige kombinasjoner av vekt og volum. Dette ville kreve betydelig computertid, mens beregning av de duale evaluatorer kun bruker brøkdeler av et sekund.

V. Lastgenerering.

Den modifiserte simplex-rutine som brukes er kun en deterministisk metode til å velge ut de av oppgitte mengder som gir størst nettofortjeneste pr. dag pr. båt.

Nå er imidlertid lasttilstrømningen stokastisk, og vi kan tenke oss at tilstrømningen av hver vare kan karakteriseres ved forventning og spredning. Disse tall kan basere seg på historiske data, eventuelt på prognoser. Dette er et problem vi ikke tar opp her, men forutsetter at slike data kan fremskaffes. I simuleringsmodellen kan man velge mellom følgende fordelinger på varetilstrømningen:

a) Man kan finne en deterministisk løsning på grunnlag av oppgitte varemengder i havnene.
b) Poissonfordeling.
c) Lognormalfordeling.
   Dette er en usymmetrisk fordeling med median under gjennomsnittet. Den gir alltid positive tall.
d) Uniform fordeling.
e) Beta-fordeling.[8]

De tre siste sannsynlighetsfordelingene genereres ved hjelp av en randomtall-generator som genererer tilfeldige tall mellom 0 og 1.

Tallene for forventning og standardavvik er oppgitt pr. dag. Vi genererer altså tilfeldige tall for varetilstrømning pr. dag og multipliserer disse med antall dager siden forrige båt var i vedkommende havn og lastet.[9]

Ved å lese inn sesongindekser for hver vare, f. eks. månedlige indekser, hvor gjennomsnittet tilsvarer 100, vil man ofte få en mer realistisk modell.

VI. Simuleringen.

Simuleringen kan grovt beskrives som følger.[10] Først leser vi inn karakteristika om alle varer, alle havner

[5] Under forutsetning av at okningen ikke medfører basisskifte i den optimale løsning.

[6] Se f. eks. A. Charnes and W. W. Cooper, «Management Models and Industrial Applications of Linear Programming», Volume I, John Wiley and Sons, New York (1961).

[7] Se Appendiks 1b.

[8] Denne brukes på samme måte som beskrevet i T. Hansen og F. Kydland: «Noen kritiske kommentarer til PERT». Bedriftsøkonomen nr. 5, 1969.

[9] I virkeligheten bruker vi en tilnærmelse for dette tidsrom, noe vi kommer inn på i avsnitt VI.

[10] Se blokkdiagram i Appendiks 2.

besøksrekkefølge. For hver besøksrekkefølge startes simuleringen med et visst antall båter i ruten, f. eks. en eller to.

Med dette antall båter gjøres så mange simuleringer som vi har bestemt på forhånd (en simulering tilsvarer at hver båt har gjort en hel rundtur). Til slutt beregnes gjennomsnitt og standardavvik for nettofortjeneste pr. dag for alle simuleringene, og også gjennomsnittlig rundturtid. Av de duale evaluatorer for hver simulering kan vi beregne hvilken gjennomsnittlig økning i nettofortjenesten pr. døgn vi ville fått ved økning av dødvekt og/eller volum.

Så økes antall båter med 1, og det samme gjentas, fremdeles med den oppgitte besøksrekkefølge. Den nye gjennomsnittlige nettofortjeneste pr. dag sammenlignes med det vi fikk med en båt mindre. Hvis fortjenesten har øket, økes antall båter med 1. Simuleringen stanser når gjennomsnittlig fortjeneste reduseres, og det optimale antall båter ved den aktuelle besøksrekkefølge er altså en mindre enn det sist undersøkte antall.

Vi leser så inn nye besøksrekkefølger og gjentar samme operasjon som ovenfor for hver besøksrekkefølge, og hver gang sammenligner vi nettofortjeneste for det optimale antall båter med det beste vi har oppnådd tidligere. Til slutt vil vi stå igjen med en optimal besøksrekkefølge, og et optimalt antall båter for denne, samt den informasjon vi fikk fra de duale evaluatorer for dette tilfelle.

Et spesielt problem er hva man skal gjøre med varer som ikke medtas ved optimaliseringen. I modellen har vi mer eller mindre tilfeldig gjort det slik at tiloversblevne varer blir liggende til neste båt kommer, men hvis de heller ikke da kommer med, går de ut av modellen (f. eks. blir fraktet på annen måte).

Et annet problem er hvor mange dager man skal bruke som grunnlag for generering av varetilstrømning. Det riktige måtte være å regne antall døgn fra forrige båt forlot havnen til sist ankomne båt forlater den. Nå optimaliseres imidlertid lasten som om man, idet man skal begynne å laste i utgangshavnen, har full informasjon om lastmengder i alle havnene for kommende rundtur. Tiden frem til man forlater hver havn er altså et resultat av optimaliseringen, mens optimaliseringen påvirkes av hvor mange dager lasten er generert. Denne vekselvirkningen er det vanskelig å få tatt hensyn til. I modellen gjør vi følgende: Når det kun går en båt i ruten, genereres varetilstrømningen i det antall døgn man brukte på forrige rundtur. Tilsvarende bruker vi, når vi har mer enn en båt, tiden fra forrige båt skulle til å laste i utgangshavnen. Vi gjør altså den tilnærmelse at vi alltid bruker tiden som gjøres ved dette skulle være ubetydelig.

For å få simuleringene i gang, må vi spesifisere en initial rundturtid. Denne bør selvsagt basere seg på et anslag over hva man regner som sannsynlig. Skulle den imidlertid ligge litt over eller under det som blir gjennomsnittet, vil dette jevne seg ut ved flere simuleringer.

Når antall båter økes med en innen samme besøksrekkefølge, baserer man seg på gjennomsnittlig rundturtid for forrige båtantall ved første generering av varetilstrømning. Dette tall divideres med det antall båter vi skal bruke.

På grunn av at den initiale rundturtid kan være nokså tilfeldig, regner vi ikke med denne ved beregning av gjennomsnitt og standardavvik for fortjeneste og rundturtider.

VII. Kontroll av scheduling.

Når vi bruker mer enn en båt, oppstår problemet med scheduling av båtene. Siden vi bruker det antall dager som er gått fra forrige båt begynte å laste i utgangshavnen som grunnlag for generering av varetilstrømning, vil et tilfeldig utslag i en eller annen retning kunne medføre at en båt begynner å kjøre innpå den forrige, og jo lenger innpå den kjører, jo mindre varer blir generert for den, og dette igjen fører til at den kjører enda mer innpå båten foran. Dette har vi forsøkt å kontrollere ved hjelp av en klokke som fortløpende registrerer tiden. Klokken føres ajour hver gang båten skal begynne å laste i utgangshavnen. Følgende variable brukes:

Løpende tid for båt nr. i (klokke)	Cl_i
Tidsgrunnlag for varegenerering for båt nr. i	t_i
Siste rundturtid	T
Øvre beskrankning på rundturtid for båten som skal til å laste i utgangshavnen (båt nr. $i+1$)	$T_{u,\,i+1}$
Tilsvarende nedre beskrankning	$T_{L,\,i+1}$
Tillatt avvik «i fase» mellom båtene	D
Antall båter	n

Avviket D vil måtte variere med antall båter, idet man får større krav til å holde båtene «i fase» med hverandre når antallet øker. I modellen gjøres dette ved å dividere en initial verdi D_{tn} med antall båter. Hvis f. eks. $D_{tn} = 10$ døgn, får vi:

Antall båter	D
2	5 døgn
3	3 1/3 døgn
4	2 1/2 døgn

Etter å ha losset båt nr. i i utgangshavnen, beregnes følgende nye verdier for ovenstående variable:

$$
\begin{aligned}
Cl_i &= Cl_i + T\\
t_i &= Cl_i - Cl_{i-1}\\
T_{u,i+1} &= Cl_i - Cl_{i+1} + t_i + D\\
T_{L,i-1} &+ T_{u,\,i+1} - 2\,D
\end{aligned}
$$

For $i = 1$ erstattes $i - 1$ med n.
For $i = n$ erstattes $i + 1$ med 1.

Denne kontroll kan føre til at i enkelte tilfeller vil en båt bli liggende og vente en stund ekstra i utgangshavnen for at den ikke skal komme for nær innpå foregående båt, men kun hvis der ikke fantes tilstrekkelige kvanta av varer på siste rundtur som kunne øke nettofortjenesten pr. dag når man tar hensyn til ventetiden hvis varene ikke tas med. Båtene vil altså i enkelte tilfeller ta med varer som de ikke ville tatt med ut fra kriteriet om maksimal fortjeneste pr. dag hvis de ikke behøvde å vente ekstra i utgangshavnen.

Simuleringsteknisk gjøres dette ved at man før man går inn i den modifiserte simplexrutine bruker verdien T_L istedenfor uttrykket $\beta_0 + \beta_1 x_1 + \ldots.$ i nevneren i uttrykket for P inntil denne sum overstiger T_L. Denne modifisering medfører altså at tvilsomme varer lettere vil komme med hvis der er knapt med varer.

I andre tilfeller kan en båt måtte la ligge igjen last som den ellers ville tatt med. Dette skjer fordi den må ha losset i utgangshavnen innen en bestemt tid for ikke å komme for langt etter båten foran.

Dette kontrolleres av en ekstra beskrankning:

$$\beta_1 x_1 + \beta_2 x_2 + \ldots. + \beta_n x_n \leqq T_u - \beta_0.$$

Det er mulig at man uten å miste noe særlig i pålitelighet kunne gjøre den forenkling at man tok utgangspunkt i en av båtene og alltid genererte varetilstrømning for den tid denne båt brukte på hver rundtur, og så dividerte varetilstrømningen for hver vare med antall båter. Da ville alle båtene ha samme lastsammensetning, og man ville spare meget i simuleringstid. Scheduling ville selvsagt ikke lenger være noe problem. Slik vi har gjort det nå, kan vi imidlertid inkorporere i modellen muligheten for å la en båt hoppe over en havn hvor der er lite last og la lasten ligge til neste båt. Ellers ville båtene alltid måtte gå innom hver havn uansett hvor lite last der var.

VIII. Prioriterte laster.

For noen ruter kan det være slik at man må ta med lasttyper selv om de kan være mindre lønnsomme enn andre tilgjengelige laster. Dette kan bero på konsesjonen eller spesielle avtaler. I modellen leser man for hver vare inn en parameter som sier om varen er prioritert eller ikke. Man vil så laste prioriterte varer først, og siden velge ut de mest lønnsomme av de resterende ved hjelp av den modifiserte simplex-rutine. Dette betyr rent regneteknisk at idet man går inn i simplexrutinen, er der skjedd følgende forandringer med beskrankningene: $w_i = 0$ for de prioriterte varer. K, V, V_1 osv. er redusert med den del av kapasiteten som går med til de prioriterte varer. Dessuten reduseres $T_u - \beta_0$ for den siste beskrankningen tilsvarende (se ovenfor).

IX. Anvendelser av modellen.

Der er i modellen flere størrelser som man kan tenkes å ville optimalisere. Vi har nevnt besøksrekkefølge og antall skip, men for at disse skal kunne optimaliseres må man på forhånd ha spesifisert båttype. Det kan også tenkes at man vil teste resultatet av alternative båttyper hva angår dødvekt, lastvolum, eventuelle spesiallasterom (f. eks. kjølerom), laste- og losseutstyr (f. eks. Munch-loaders), servicehastighet etc.

Man kan også simulere seg frem til en mer optimal utnyttelse av de skip man allerede har i rute. For å muliggjøre dette, kan vi i modellen samtidig kjøre med flere båter som har forskjellige karakteristika.

I linjefart må man fastlegge et ruteopplegg slik at båtene ankommer til nokså faste tider og med jevne mellomrom. Dette ruteopplegg vil i modellen komme som et biprodukt for den optimale kombinasjon. Man kan også foreta simulering hvor båtene alltid skal bruke en på forhånd fastlagt tid pr. rundtur, og lastene vil da optimaliseres ut fra dette. Simuleringsteknisk låses ruteopplegget ved at man setter $D_{in} = 0$ for kontroll av scheduling (se avsnitt VII).

Til slutt kan vi nevne muligheten for å bruke modellen i driften av en rute. Da vil alle skipsdata være gitt. På bestemte punkter i ruten vet man gjerne med sikkerhet lastmengdene i den nærmeste, eventuelt de nærmeste havnene, mens lastmengdene videre i ruten sannsynligvis er mer usikre jo lenger borte havnene ligger. På grunnlag av gitte mengder av noen varer og sannsynlighetsfordelinger for de usikre varene, kunne man simulere seg frem til hvilke varer som bør lastes, eventuelt også om besøksrekkefølgen bør endres litt hvis der er mulighet for det.

Siden en modell er en abstrahering av virkeligheten, må den nødvendigvis være ufullstendig og unøyaktig på forskjellige måter. Virkeligheten er altfor kompleks til at man kan analysere den skikkelig uten den forenkling som modeller representerer. Det virkelige grunnlag for å bedømme en modells nytte er ikke om den er fullstendig nøyaktig, men heller om den inneholder nok relevante elementer til at den kan bli effektivt anvendt av bedriftsledere til å forbedre deres resultater. Vi tror at denne modell kan være brukbar i så måte.

I arbeidet med den modell som her er beskrevet, ble det lagt vekt på å gjøre den så generell at den kunne anvendes på forskjelligartede problemer i linjefarten. Når selve rammen nå er bygget, kan man siden bygge inn spesielle aspekter av virkeligheten som den enkelte bruker måtte ønske å ta i betraktning.

Appendiks 1.

I dette Appendix skal vi kort forklare prinsippet bak løsningen av et maksimeringsproblem med lineære beskrankninger, hvor objektfunksjonen er ikke-lineær og består av en brøk hvor både teller og nevner er lineære i de variable.[11] Til slutt skal vi regne gjennom et numerisk eksempel. Vi forutsetter i det følgende at simplex-metoden er kjent.

a) *Det primære problem.*

Vårt problem kan formuleres:[12]

$$(1) \qquad \text{Maks } P = \frac{\alpha_0 + \alpha_1 x_1 + \ldots\ldots + \alpha_n x_n}{\beta_0 + \beta_1 x_1 + \ldots\ldots + \beta_n x_n}$$

med beskrankningene:

$$(2) \qquad \begin{array}{c} a_{11}x_1 + \ldots\ldots + a_{1n}x_n \leq b_1 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{m1}x_1 + \ldots\ldots + a_{mn}x_n \leq b_m \\ x_i \geq 0,\ i = 1, \ldots, n \end{array}$$

Vi innfører slackvariable $x_{n+1}, \ldots, x_{n+m}$, og får at (2) er ekvivalent med:

$$(3) \qquad \begin{array}{c} a_{11}x_1 + \ldots\ldots + a_{1n}x_n + x_{n+1} = b_1 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{m1}x_1 + \ldots\ldots + a_{mn}x_n + x_{n+m} = b_m \\ x_i \geq 0,\ i = 1, \ldots, n + m \end{array}$$

Objektfunksjonen kan omformuleres til:

$$(4) \quad (\alpha_0 - \beta_0 P) + (\alpha_1 - \beta_1 P)x_1 + \ldots\ldots + (\alpha_n - \beta_n P)x_n = 0$$

Vi skal altså maksimere P slik at (4) er tilfredsstilt.

Vi vil gjøre enda en omformulering av vårt problem, idet vi tenker oss at vi vil maksimere

$$f = \Sigma \; (\alpha_j - \beta_j \overline{P})x_j$$

Da er problemet overført til et generelt problem innenfor parametrisk programmering, og det kan løses ved hjelp av simplex-metoden.

Vår initiale basis består av kolonnevektorene svarende til slackvariablene $x_{n+1}, \ldots, x_{n+m}$. Vårt første simplextableau er vist i tabell 1.

$$z_j = \sum_{i=1}^{m} C_\beta^i \, a_{ij}, \; j = 1, \ldots, n + m$$

c_B \ c_j	$\alpha_1 - \beta_1\overline{P}$	$\alpha_2 - \beta_2\overline{P}$	......	$\alpha_n - \beta_n\overline{P}$	0	0	......	0	
	x_1	x_2	......	x_n	x_{n+1}	x_{n+2}	......	x_{n+m}	b
0	a_{11}	a_{12}	......	a_{1n}	1	0	......	0	b_1
0	a_{21}	a_{22}	......	a_{2n}	0	1	......	0	b_2
.	......	......	......	......	......	......	......	......	.
.	......	......	......	......	......	......	......	......	.
.	......	......	......	......	......	......	......	......	.
0	a_{m1}	a_{m2}	......	a_{mn}	0	0	......	1	b_m
z_j	0	0	......	0	0	0	......	0	
$z_j - c_j$	$\beta_1\overline{P} - \alpha_1$	$\beta_2\overline{P} - \alpha_2$	......	$\beta_n\overline{P} - \alpha_n$	0	0	......	0	

Tabell 1.

Med en mulig løsning skal vi mene en løsning som tilfredsstiller (3). Vektoren $x = (x_1, \ldots, x_n, x_{n+1}, \ldots, x_{n+m}) = (0, \ldots, 0, b_1, \ldots, b_m)$ er altså en mulig løsning. For at den skal være en optimal løsning, må vi ha $z_j - c_j \geq 0$ for alle j. Derav følger at $\beta_j\overline{P} - \alpha_j \geq 0$ for alle $j \Rightarrow \overline{P} \geq \dfrac{\alpha_j}{\beta_j}$ for alle j. La K_1 være $\underset{j}{\text{maks}} \left\{\dfrac{\alpha_j}{\beta_j}\right\}$. Betingelsen for at $x = (0, \ldots, 0, b_1, \ldots, b_m)$ skal være en optimal løsning, er altså at $\overline{P} \geq K_1$. Vi beregner $P = \dfrac{\alpha_0}{\beta_0}$. Hvis $P \geq K_1$, har vi funnet den vektor x som gjør P maksimal. I motsatt fall, dvs. hvis $P < K_1$, er løsningen ikke optimal.

La $M = \{j_1, \ldots, j_m\}$ = mengden av indekser for basisvariablene. Ved å sette $\overline{P} = K_1$, vil vi ha minst en j som ikke er med i M slik at $z_j - c_j = 0$. Anta at denne indeks er k. Vi velger x_k som ny basisvariabel

[11] Se også Pollak, Novaes og Frankel: «On the Optimization and Integration of Shipping Ventures». International Shipbuilding Progress, July 1965.

[12] Denne problemformulering går under betegnelsen fractional programming.

tredsstilt for, kan vi velge den med lavest indeks).

Vi finner ut hvilken vektor som skal byttes ut og beregner nytt simplextableau på vanlig måte. Dette gir oss et nytt sett av $z_j - c_j = \alpha_j' - \beta_j'\overline{P}$ og en ny vektor x som er en optimal løsning dersom $z_j - c_j \geq 0$ for alle j. Dette krav vil være oppfylt dersom $K_2 \leq \overline{P} \leq K_1$, hvor K_2 er en ny konstant beregnet på tilsvarende måte som K_1. Vi beregner så

$$P = \frac{\alpha_0 + \alpha_k x_k}{\beta_0 + \beta_k x_k}$$

Dersom $K_2 \leq P \leq K_1$, er P maksimal. I motsatt fall, dvs. dersom $P < K_2$, settes $\overline{P} = K_2$, og vi finner en ny basisvariabel.

Etter neste iterasjon får vi som betingelse for optimal løsning at $K_3 \leq \overline{P} \leq K_2$, hvor K_3 er en konstant. Dersom P ligger innenfor dette intervall, har vi funnet maksimum for P. I motsatt fall fortsetter vi på samme måte inntil P ligger innenfor det intervall som gjør alle $z_j - c_j \geq 0$.

For å illustrere denne metoden, skal vi regne gjennom et lite numerisk eksempel. Vi tenker oss at vi har følgende problem:

$$\text{Maks } P = \frac{-20 + 5x_1 + 3x_2 + 2x_3}{4 + 2x_1 + 3x_2 + x_3}$$

med beskrankningene:

$$\begin{aligned} x_1 \qquad\qquad &\leq 10 \\ x_2 \qquad &\leq 8 \\ x_3 &\leq 6 \\ x_1 + x_2 + x_3 &\leq 18 \\ x_1 \qquad + 2x_3 &\leq 18 \\ x_i \geq 0,\ i &= 1,2,3 \end{aligned}$$

Objektfunksjonen kan omformes til:

Maks P slik at

$(-20 - 4P) + (5 - 2P)x_1 + (3 - 3P)x_2 + (2 - P)x_3 = 0$

Etter innføring av slackvariable $x_4, \ldots, x_8$, kan vi sette opp vårt initiale simplextableau hvor vi tenker oss at vi skal maksimalisere

$$f = (5 - 2\overline{P})x_1 + (3 - 3\overline{P})x_2 + (2 - \overline{P})x_3 :$$

c_j	$5 - 2\overline{P}$	$3 - 3\overline{P}$	$2 - \overline{P}$	0	0	0	0	0	
c_B	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	b
0	1	0	0	1	0	0	0	0	10
0	0	1	0	0	1	0	0	0	8
0	0	0	1	0	0	1	0	0	6
0	1	1	1	0	0	0	1	0	18
0	1	0	2	0	0	0	0	1	18
z_j	0	0	0	0	0	0	0	0	
$z_j - c_j$	$2\overline{P} - 5$	$3\overline{P} - 3$	$\overline{P} - 2$	0	0	0	0	0	

Tabell 2.

om denne løsning gjør P maksimal. Kravet $z_j - c_j \geq 0$ gir:

$$\left.\begin{aligned} 2\overline{P} - 5 \geq 0 &\Rightarrow \overline{P} \geq 2{,}5 \\ 3\overline{P} - 3 \geq 0 &\Rightarrow \overline{P} \geq 1 \\ \overline{P} - 2 \geq 0 &\Rightarrow \overline{P} \geq 2 \end{aligned}\right\} \Rightarrow \overline{P} \geq 2{,}5$$

$$P = \frac{-20}{4} = -5, \text{ dvs. løsningen er ikke optimal.}$$

Hvis vi setter $\overline{P} = 2{,}5$, blir $z_1 - c_1 = 0$, og x_1 er vår nye basisvariabel.

Vi beregner nytt simplextableau på vanlig måte:

c_j	$5 - 2\overline{P}$	$3 - 3\overline{P}$	$2 - \overline{P}$	0	0	0	0	0	
c_B	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	b
$5 - 2\overline{P}$	1	0	0	1	0	0	0	0	10
0	0	1	0	0	1	0	0	0	8
0	0	0	1	0	0	1	0	0	6
0	0	1	1	-1	0	0	1	0	8
0	0	0	2	-1	0	0	0	1	8
z_j	$5 - 2\overline{P}$	0	0	$5 - 2\overline{P}$	0	0	0	0	
$z_j - c_j$	0	$3\overline{P} - 3$	$\overline{P} - 2$	$5 - 2\overline{P}$	0	0	0	0	

Tabell 3.

Vår nye mulige løsning gir $x_1 = 10$, $x_2 = 0$, $x_3 = 0$. Kravet $z_j - c_j \geq 0$ impliserer:

$$\left.\begin{aligned} 3\overline{P} - 3 \quad \geq 0 &\Rightarrow \overline{P} \geq 1 \\ \overline{P} - 2 \quad \geq 0 &\Rightarrow \overline{P} \geq 2 \\ 5 \quad - 2\overline{P} \geq 0 &\Rightarrow \overline{P} \leq 2{,}5 \end{aligned}\right\} \Rightarrow 2 \leq \overline{P} \leq 2{,}5$$

$$P = \frac{-20 + 5 \,.\, 10}{4 + 2 \,.\, 10} = 1{,}25,$$

og siden vi har $P < 2$, er heller ikke denne P maksimal.

Vi innfører x_3 som ny basisvariabel, se tabell 4.

Vi får da:

$$x_1 = 10,\ x_2 = 0,\ x_3 = 4.$$

Kravet $z_j - c_j \geq 0$ gir:

$$\left.\begin{aligned} 3\overline{P} - 3 \qquad &\geq 0 \Rightarrow \overline{P} \geq 1 \\ 4 \quad - 1{,}5\overline{P} &\geq 0 \Rightarrow \overline{P} \leq 2{,}67 \\ 1 \quad - 0{,}5\overline{P} &\geq 0 \Rightarrow \overline{P} \leq 2 \end{aligned}\right\} \Rightarrow 1 \leq \overline{P} \leq 2$$

$$P = \frac{-20 + 50 + 8}{4 + 20 + 4} = \frac{38}{28} = 1{,}3571,$$

dvs. $1 \leq P \leq 2$, som betyr at P har oppnådd maksimum.

Vi ser at dersom konstantleddet i telleren i uttrykket for P hadde vært lavere enn -30, ville vi hatt $P < 1$, og det ville da vært optimalt også å innføre x_2 som basisvariabel.

c_B \ c_j	$5-2\overline{P}$	$3-3\overline{P}$	$2-\overline{P}$	0	0	0	0	0	
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	b
$5-2\overline{P}$	1	0	0	1	0	0	0	0	10
0	0	1	0	0	1	0	0	0	8
0	0	0	0	$\frac{1}{2}$	0	1	0	$-\frac{1}{2}$	2
0	0	1	0	$-\frac{1}{2}$	0	0	1	$-\frac{1}{2}$	4
$2-\overline{P}$	0	0	1	$-\frac{1}{2}$	0	0	0	$\frac{1}{2}$	4
z_j	$5-2\overline{P}$	0	$2-\overline{P}$	$4-1.5\overline{P}$	0	0	0	$1-0.5\overline{P}$	
z_j-c_j	0	$3\overline{P}-3$	0	$4-1.5\overline{P}$	0	0	0	$1-0.5\overline{P}$	

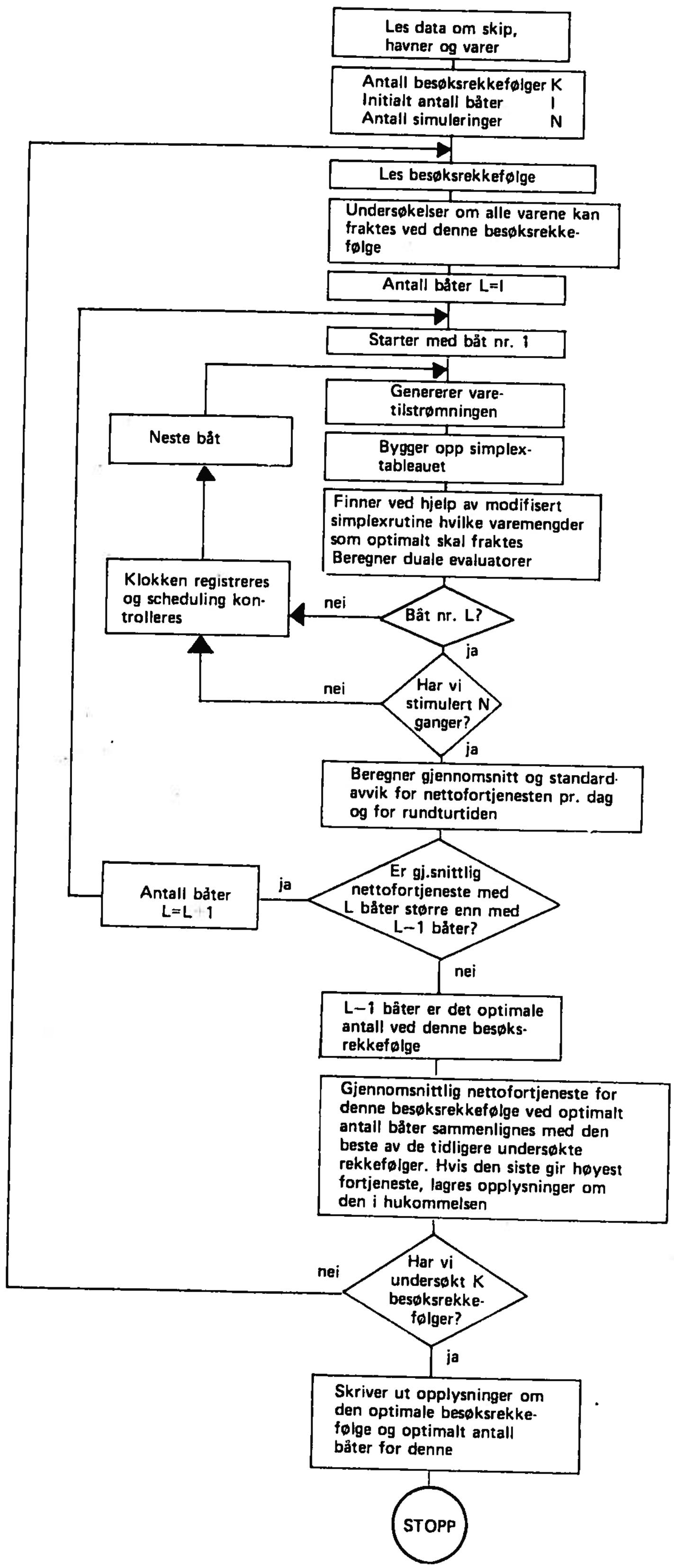
Tabell 4.

b) *Duale evaluatorer.*

Som kjent vil man i det optimale simplextableau for et lineært programmeringsproblem finne den duale evaluator for beskrankning nr. i som størrelsen på $z_{n+i}-c_{n+i}$, dvs. i kolonnen tilsvarende slackvariabelen for beskrankning nr. i. Den duale evaluator for en beskrankning vil bare kunne være positiv dersom den tilsvarende slackvariabel er lik 0, dvs. at ulikheten i den opprinnelige formulering av beskrankningen holder som likhet, og den duale evaluator sier hvor meget objektfunksjonen ville øke pr. enhet økning i vedkommende beskrankning b_i under forutsetning av at økningen ikke medfører basisskifte i den optimale løsning.

Mens hver z_j-c_j i det lineære tilfelle kun består av et element, i det initiale tableau $-c_j$, vil våre

Blokkdiagram over hovedpunktene i simuleringsmodellen.



alene og en foran P, i det initiale tableau $-\alpha_j + \beta_j\overline{P}$. Under beregningene i simplextableauet blir imidlertid disse koeffisientene α_j og β_j forandret på nøyaktig samme måte som de ville blitt dersom telleren alene eller nevneren alene var objektfunksjon. Det kan vises at de tjener som duale evaluatorer for henholdsvis teller og nevner hver for seg, selv om vårt optimale tableau generelt ikke ville være optimalt med kun telleren eller nevneren som objektfunksjon.

Dersom vi i det optimale tableau skriver $z_{n+i} - c_{n+i} = \alpha_i' - \beta_i'\overline{P}$, sier altså α_i' i følge vår påstand ovenfor hvor meget telleren ville øke pr. enhet økning i b_i, mens β_i' sier hvor meget nevneren ville øke. Dersom vi betegner telleren i vår optimale løsning med T og nevneren med N, kan vi finne hvor meget vår objektfunksjon P ville øke ved en økning av b_i med k enheter, ved å beregne

$$\frac{T + k\alpha_i'}{N + k\beta_i'} - \frac{T}{N}, \quad i = 1, \ldots, m.$$

Hvis vi går tilbake til vårt numeriske eksempel, tabell 4, ser vi at to duale evaluatorer er forskjellig fra 0, nemlig tilsvarende beskrankning nr. 1 og nr. 5. Den første sier at en økning av b_1 med en enhet fra 10 til 11 enheter ville føre til en økning av P på:

$$\frac{38 + 4}{28 + 1.5} - \frac{38}{28} = 0.06659.$$

Hvis samtidig b_5 øket fra 18 til 19 enheter, ville P øke med

$$\frac{38 + 4 + 1}{28 + 1.5 + 0.5} - \frac{38}{28} = 0.07619.$$

RETTELSE

I «Diskontopolitikkens rolle i den internasjonale pengehusholdning», Sosialøkonomen nr. 8, 1969, side 7, 1. spalte, avsnittet *Renteparitetsteorien,* 5.—7. linje ovenfra, står:

«Samtidig blir det en økt etterspørsel etter statskasseveksler i Land A og en redusert etterspørsel etter statskasseveksler i Land B.»

Uttrykkene Land A og Land B skal bytte plass.