# OPERATIONSANALYSE - EKSAMENSNOTER



Tal i parentes refererer til de relevante sider i Winston.

Fejl, mangler, udeladelser og unøjagtigheder i disse noter er ingen undskyldning for en dålig eksamenskarakter, og undertegnede kan på ingen måde stilles til regnskab.

# Kapitel 4

**Konvertering til standardform** (124-125)

Tilføj en slack-variabel $s_i$ til alle ≤-bibetingelser og en (minus) excess-variabel $e_i$ til alle ≥-bibetingelser.

**Løsning af LP-problemer via simplex**

<u>LP-problemer på normalform, dvs. alle bibetingelser er ≤-bibetingelser (standard simplex) (133-140):</u>

1. Konvertér problemet til standardform.
2. Omform objektfunktion så højresiden er lig 0.
3. Er der tale om et maximerings- (minimerings)problem?
4. Find pivotsøjlen, dvs. den søjle med det største negative (positive) element i række 0.
5. Lav ratios for hvert element i pivotsøjlen (minus række 0) ved at dividere elementet i pivotsøjlen op i rækkens højreside-element.
6. Find pivotelement, dvs. det element i pivotsøjlen med den laveste positive ratio.
7. Bring pivotvariablen ind i basis, dvs. omform pivotelementet til et 1-tal og skab 0'er over og under dette.
8. Er den nuværende løsning optimal, dvs. er alle elementer i række 0 ikke-negative (ikke-positive)? Hvis ikke – gå videre fra punkt 4.
9. Når der haves en optimal løsning kan denne aflæses i "højre-side"-søjlen.

<u>Er der tale om et LP-problem, hvor alle bibetingelser er ≥-bibetingelser (268-):</u>

1. Løs det duale problem, og aflæs løsningen til det primære problem i række 0.

<u>Er der tale om et LP-problem, hvor mindst 2 af mulighederne ≤, ≥, = indgår (164-174):</u>

1. Brug Big M metoden.
2. Konvertér til standardform og tilføj artificiale variable $a_i$ til ≥ og =-bibetingelser.
3. Ved maksimering (minimering) tillæg (fratræk) M gange samtlige artificiale variable i objektfunktionen (før omformning), hvor M er et meget stort tal.
4. Eliminér de artificiale variable fra række 0 ved rækkeoperationer.
5. Fortsæt nu med standard-simplex.

**Tilføjelser til simplex**

Har en ikke-basis-variabel i det optimale tableau er 0 i række 0, er den fundne løsning ikke entydig. En anden optimal løsning kan da findes ved bringe den givne variabel ind i basis (146-147).

Hvis den valgte pivotrække udelukkende har negative elementer i bibetingelserne, siges problemet at være ubundet (z kan antage vilkårligt store/små værdier). Der er ofte i sådanne tilfælde noget galt (148-150).

Hvis en artificial variabel indgår i den optimale løsning i et Big M problem, har problemet ingen løsninger (169).

# Kapital 6

**Sensivitetsanalyser**

Ændring i optimal løsning ved ændring LP-koefficenterne.m – m

Formler (244):

- $x_j$'s række 0-koefficent i det optimale tableau = $c_j' = c_{BV} B^{-1}a_j - c_j$.
- Højre sider i det optimale tableau = $b' = B^{-1}b$.
- $x_j$'s bibetingelsessøjle = $B^{-1}a_j$.
- z-værdi i optimalt tableau = $c_{BV} B^{-1}b$.
- $s_i$'s række 0-koefficent i det optimale tableau = i'te element i $c_{BV} B^{-1}$.
- $e_i$'s række 0-koefficent i det optimale tableau = - (i'te element i $c_{BV} B^{-1}$).
- $a_i$'s række 0-koefficent i det optimale tableau = (i'te element i $c_{BV} B^{-1}$) $\pm$ M (+ ved maximering, – ved minimering).
- Matricen $B^{-1}$ kan findes direkte i det optimale tableau som søjlerne (minus række 0) under slack- og de artificiale variable – i den i tableauet givne rækkefølge.

Ændring af række 0-koefficenten for en ikke-basis-variabel (249-250):

1. Konstruér ny optimal række 0-koefficent via formlen $c_j' = c_{BV} B^{-1}a_j - c_j$.

2. Er der tale om maksimering (minimering) vil den fundne basis stadig være optimal, hvis $c_j' \geq 0$ ($c_j' \leq 0$).
3. Hvis den fundne basis ikke længere er optimal, indsættes $c_j'$ i det ”optimale” tableau og der køres simplex indtil en ny optimal løsning er fundet.

Ændring af række 0-koefficenten for en basis-variabel (250-253):

1. Bestem $c_{BV}$ $B^{-1}$-vektoren.
2. Konstruér nye optimale række 0-koefficenter for alle ikke-basis-variable via formlen $c_j' = c_{BV} B^{-1}a_j - c_j$. Koefficenterne for $s_i$, $e_i$ og $a_i$ bestemmes direkte via $c_{BV}$ $B^{-1}$.
3. Hvis der er tale om maksimering (minimering), vil basis forblive optimal, hvis de beregnede koefficienter alle er ikke-negative (ikke-positive). Bemærk dog at z ændrer sig!
4. Hvis basis ikke længere er optimal, indsættes de fundne koefficienter samt den nye z-værdi i det ”optimale” tableau og der køres simplex indtil en ny optimal løsning er fundet.

Ændring af en højre-side (254-257):

1. Konstruér nye højre-sider via formlen $b' = B^{-1}b$.
2. Hvis alle elementer i b’ er ikke-negative, forbliver basis optimal. Bemærk dog at z samt basis-variablenes værdier ændrer sig!
3. Hvis ovenstående ikke er tilfældet, indsættes b’ og den nye z-værdi i det ”optimale” tableau, og der køres dual-simplex indtil en ny optimal løsning er fundet.

Ændring af søjlen for en ikke-basis-variabel (258-259):

1. Konstruér ny optimal række 0-koefficent via formlen $c_j' = c_{BV} B^{-1}a_j - c_j$.
2. Er der tale om maksimering (minimering) vil den fundne basis stadig være optimal, hvis $c_j' \geq 0$ ($c_j' \leq 0$).
3. Hvis ovenstående ikke er tilfældet, beregnes variablens bibetingelsessøjle som $B^{-1}a_j$. Denne samt $c_j'$ indsættes i det ”optimale tableau og der køres simplex indtil en optimal løsning er fundet.

Tilføjelse af ny aktivitet (259-260):

1. Konstruér optimal række 0-koefficent for den nye variabel via formlen $c_j' = c_{BV} B^{-1}a_j' - c_j$.

2. Er der tale om maksimering (minimering) vil den fundne basis stadig være optimal, hvis $c_j' \geq 0$ ($c_j' \leq 0$).
3. Hvis ovenstående ikke er tilfældet, beregnes variablens bibetingelsessøjle som $B^{-1}a_j$. Denne samt $c_j'$ indsættes i det "optimale tableau og der køres simplex indtil en optimal løsning er fundet.

**Dualitet**

Konstruktion af det duale problem (268-274):

1. Kontrollér at problemet er på normalform.
2. Er problemet ikke på normalform ændres ≥-bibetingelser til ≤-bibetingelser ved at gange igennem med –1, og =-bibetingelser ændres til en ≤- og en ≥-bibetingelse.
3. Problemet ændres fra maksimering til minimering (og omvendt).
4. Højre-siderne bliver nye koefficenter i række 0, mens de oprindelige koefficenter i række 0 bliver nye højre-sider.
5. Variablene ændrer navn fra $x_1,\ldots,x_n$ til $y_n,\ldots,y_m$.
6. Bibetingelse j's venstre-side-koefficienter bliver koefficienter i søjle j (minus række 0).
7. Hvis der i det primære problem var n variable og m bibetingelser, er det nu omvendt.
8. Nu kan problemet konverteres til standardform.

Aflæsning af den duale løsning i det primære optimale tableau (285-286):

- Optimal $x_i/y_i$ ved ≤-bibetingelse = $s_i$-koefficent i række 0
- Optimal $x_i/y_i$ ved ≥-bibetingelse = – ($e_i$-koefficent i række 0)
- Optimal $x_i/y_i$ ved =-bibetingelse = ($a_i$-koefficent i række 0) – M (+M ved min-problem)

Den optimale værdi af objekt-fuktionen er ens i både det primære og det duale problem.

Implikationer af dualitet(288-289, 301-303):

De duale variable svarer til skyggepriserne i det primære problem, dvs. hvor meget øges objektfunktionen, hvis en af højresiderne øges med 1 enhed.

Er en bibetingelse ikke bindende, er den tilhørende duale variabel 0.

De duale variable kan ofte tolkes som priser på de tilhørende ressourcer.

<u>Dual-simplex (305-310):</u>

1. Bruges til løsning af problemer hvor alle koefficienter i række 0 er ikke-negative, og der eksisterer mindst én negativ højre-side.
2. Som pivotrække vælges rækken med den mest negative højre side.
3. Lav ratios for hvert negative element i pivotrækken (minus 1. søjle) ved at dividere elementet i pivotrækken op i søjlens element i række 0.
4. Find pivotelement, dvs. det element i pivotrækken med den laveste numeriske ratio.
5. Bring pivotsøjlens variabel ind i basis via *række*operationer.
6. Når der ikke er flere negative højresider, er den optimale løsning fundet.

Dual-simplex bruges primært i følgende situationer:

- Hvis der tilføjes en ny bibetingelse, som den fundne optimale løsning ikke opfylder.
- Hvis højre-siden ændres, så højre-siderne i det optimale tableau ikke alle længere er positive.
- Til løsning af minimeringsproblemer på¨normalform.

## Kapitel 7

### Transport-problemer

Et transport-problem er balanceret, hvis det samlede udbud er lig den samlede efterspørgsel. Er dette ikke tilfældet balanceres problemet ved at indføre en dummy-størrelse, der således står for det manglende udbud/efterspørgsel. Transportomkostninger til/fra dummyen sættes til 0 (340-342). Indledningsvis findes en mulig basis-løsning på én følgende af måder.

<u>Minimum-cost metoden (356-358):</u>

1. Balancér evt. problemet og opstil transport-tableauet.
2. Kontrollér at der er tale om et minimeringsproblem. Hvis ikke fratrækkes hver omkostning fra den største, hvorved nye omkostninger fremkommer.
3. Find cellen (minus ”dummy-celler”) med lavest omkostninger og gør denne så stor som muligt. Rækker/søjler der er ”mættede” streges ud (*aldrig* både række og søjle, selv om begge er ”mættede”).

4. Fortsæt sådan indtil alle rækker og streget ud, idet kun ikke udstregede rækker og søjler må benyttes.

<u>Vogel's metode (358-360):</u>

1. Balancér evt. problemet og opstil transport-tableauet.
2. Kontrollér at der er tale om et minimeringsproblem. Hvis ikke fratrækkes hver omkostning fra den største, hvorved nye omkostninger fremkommer.
3. Konstruér straffe for hver række og søjle (minus ”dummy-celler”) som differencen mellem de to laveste omkostninger.
5. Find rækken/søjlen med højest straf og i denne cellen med lavest omkostninger og gør denne så stor som mulig. Rækker/søjler der er ”mættede” streges ud (*aldrig* både række og søjle, selv om begge er ”mættede”).
4. Konstruér nye straffe for ikke udstregede rækker og søjler.
5. Fortsæt sådan indtil alle rækker og streget ud, idet kun ikke udstregede rækker og søjler må benyttes.

<u>Transport-simplex (361-369):</u>

1. Find de duale variable for hver række og søjle, idet første rækkes duale variabel sættes lig 0 og de øvrige konstrueres, så omkostningerne i en basis-celle er lig summen af den respektive række og søjles duale variabel.
2. Konstruér række 0-koefficenter ($c_j$') for alle ikke-basis-celler, idet disse er lig summen af den respektive række og søjles duale variabel fratrukket cellens omkostninger.
3. Hvis alle $c_j$'er er ikke-positive, er den fundne løsning optimal.
4. Er ovenstående ikke tilfældet, indættes cellen med størst positiv $c_j$' i basis ved at konstruere et loop. Denne celle betegnes *lige*, mens de øvrige basis-variable i loop'et betegnes *ulige*, *lige*, *ulige* osv. Mindste *ulige* værdi lægges til alle lige celler og fratrækkes alle ulige, således at kun én variabel forlader basis.
5. Nu fortsættes fra pkt. 2 indtil en optimal løsning er fundet.

**Assignment-problemer**

Svarer til transportproblemer, hvor udbuddet fra hver udbyder skal leveres til kun en efterspørger, og hvor udbud og efterspørgsel i hvert punkt er lig 1.

<u>Den ungarske metode (375):</u>

1. Opstil en n $\times$ n - omkostningsmatrix.
2. Kontrollér at der er tale om minimering. I tilfælde af maksimering fratrækkes hver omkostning fra den største, hvorved nye omkostninger fremkommer.
3. Fratræk i hver række mindste værdi i rækken. Gør det samme for hver søjle.
4. Dæk samtlige nuller med mindst mulige antal vandrette og lodrette streger.
5. Hvis antallet af streger er lig n, kan den optimale løsning aflæses.
6. Hvis antallet af streger er mindre end n, fratækkes mindste udækkede værdi fra samtlige udækkede celler, mens samme værdi lægges til i samtlige dobbelt-dækkede celler.
7. Nu fortsættes fra pkt. 3 indtil en optimal løsning er fundet.

**Transhipment.problemer** (380-383)

Som transportproblemer – der tillades blot mellemstationer/transhipment-punkter.

- Hvert transhipment-punkt indgår både som udbyder og efterspørger i transport-tableauet.
- Hver transhipment-punkt får udbud/efterspørgsel lig det samlede udbud (incl. dummy).
- Transportomkostninger fra et transhipment-punkt til sig selv sættes lig 0.
- Er en transport mellem to punkter ikke mulig sættes omkostningerne til M.

## Kapitel 8

**Korteste vej – shortest path**

Problemer, hvor den korteste/billigste/hurtigste rute mellem to punkter i et netværk ønskes fundet.

<u>Dijkstra's algoritme (397-398):</u>

1. Opstil evt. netværket grafisk.
2. Giv startpunktet værdien 0.
3. Øvrige punkter kan gives en værdi, når samtlige, forudgående tilknyttede punkter har fået en værdi.
4. Øvrige punkter gives en værdi, der er et minimum af samtlige, forudgående tilknyttede punkters værdi plus den respektive vejs længde.
5. Algoritmen slutter når slutpunktet har fået en værdi. Denne værdi er den korteste vejs længde,

6. Den korteste vej findes ved at gå tilbage gennem netværket.

**Maksimal strøm – maximum flow**

Problemer, hvor den størst mulige strøm (af olie, trafik, kapital etc.) mellem to punkter i et netværk ønskes bestemt.

Ford-Fulkerson's algoritme (405-411):

1. Opstil evt. netværket grafisk.
2. Send en strøm fra start- til slutpunktet, idet man må bevæge sig forlæns af en vej, hvis den ledige kapacitet er positiv, og baglæns, hvis vejens eksisterende strøm er positiv. En strøm må ikke få nogen vejs strøm til overskride sin kapacitet eller til at blive negativ.
3. Pkt. 2 gentages, indtil det ikke er muligt at sende mere fra start- til slutpunktet.
4. Tjek at den fundne løsning er optimal ved at finde det minimale snit. Hvis kapaciteten af dette er lig den fundne strøm, er løsningen optimal.

**Kritisk vej – critical path**

Bruges til tilrettelægge projekter i tid og til at bestemme kritiske aktiviteter.

Konstruktion af netværk.(415-417):

- Hver vej repræsenterer en aktivitet, mens hvert punkt repræsenterer en tilstand. Længden af hver vej angiver aktivitetens tidsudstrækning.
- Ét punkt angiver en starttilstand, mens ét punkt angiver sluttilstanden.
- En aktivitet kan repræsenteres ved én vej.
- To punkter må kun forbindes af én vej (ellers indføres dummy-aktiviteter).

Beregning af Early Event Time (ET) (418-419):

- ET(i) beregnes for hvert punkt i, som det tidligste tidspunkt hvorpå punktets tilstand kan opnås.
- ET(i) beregnes som maksimum af ET(j) + $t_{ji}$ (for samtlige umiddelbart forudgående punkter j), hvor $t_{ji}$ er længden af vejen fra j til i.

<u>Beregning af Late Event Time (LT) (419-420):</u>

- LT(i) beregnes for hvert punkt i, som det seneste tidspunkt et punkt kan nås uden at forsinke det samlede projekt.
- LT(i) beregnes som minimum af T(j) + $t_{ij}$ (for samtlige umiddelbart efterfølgende punkter j),hvor $t_{ij}$ er længden af vejen fra i til j.

<u>Beregning af Total Float (TF) (420-421):</u>

- TF(i,j) beregnes for hver vej, som den maksimale tid som en aktivitet kan udsættes uden at forsinke det samlede projekt.
- TF(i,j) beregnes som LT(j) – ET(i) – $t_{ij}$.

<u>Bestemmelse af den kritiske vej (421):</u>

- Hver aktivitet med TF = 0 betegnes kritisk aktivitet, idet forsinkelse af en sådan aktivitet vil forsinke det samlede projekt.
- En rute fra start- til slutpunket, der udelukkende består af kritiske aktiviteter, kaldes en kritisk vej.

<u>Beregning af Free Float (TF) (422):</u>

- FF(i,j) beregnes for hver vej, som den maksimale tid en aktivitet kan udsættes uden at forsinke senere aktiviteter.
- FF(i,j) beregnes som ET(j) – ET(i) – $t_{ij}$.

**Mindst udspændte træ – minimum spanning tree** (442-444)

1. Opstil evt. netværket grafisk.
2. Tag udgangspunkt i et tilfældigt punkt.
3. Forbind dette punkt med nærmeste andet punkt.
4. Forbind det hidtil dannede ”træ” med nærmeste andet punkt.
5. Forsæt med pkt. 4 indtil alle punkter i netværket er forbundet (består netværket af n punkter, skal ”træet” have n-1 ”grene”).

# Kapitel 9

**Løsning af generelle heltals-problemer**

Problemer, hvor en eller flere variable kun må antage heltallige, ikke-negative værdier.

Løsning ved Branch-and-Bound (502-513):

1) Løs problemets *LP-relaxation*. Hvis alle beslutnings-variable er heltallige, er en optimal løsning fundet.
2) Er ovenstående ikke tilfældet vælges en ikke heltallig variabel $x_i$, hvor $a < x_i < b$ (a,b er nærmeste heltallige værdier). Nu dannes to nye underproblemer med den ekstra betingelse $x_i \leq a$ henholdsvis $x_i \geq b$.
3) Regler for dannelse af nye underproblemer:
   - Giver et underproblem en ikke-heltallig løsning dannes nye underproblemer som i pkt. 2.
   - Giver et underproblem en heltallig løsning dannes *ikke* nye underproblemer.
   - Giver et underproblem en mindre optimal z-værdi end en heltallig løsning til et andet underproblem, dannes *ikke* nye underproblemer.
   - Giver et underproblem en z-værdi, som er mindre end 1 enhed mere optimal end en heltallig løsning til et andet underproblem, dannes *ikke* nye underproblemer.
4) Når der ikke kan dannes flere nye underproblemer, findes den nye optimale løsning som den heltallige løsning til et underproblem, der har den mest optimale z-værdi.

Lav gerne et *grene-diagram*!

Løsning af heltals problemer, hvor ikke alle variable skal være heltallige (515):

- Der "branches" kun på heltalsbetingede variable.
- Der benyttes branch-and-bound som ovenfor.

Løsning ved Cutting Plane – Fractional Cut (539-542):

1. Find det optimale simplex-tableau for problemets LP-relaxation. Hvis alle variable antager heltallige værdier er den fundne løsning optimal.
2. Find den bibetingelse i tableauet, hvor højresidens brøkdel er tættest på en halv.

3. Konstruér ny bibetingelse udfra ovenstående bibetingelse. Bibetingelsen bliver:
   - 1 × (sum af brøkdele på venstresiden) ≤ - 1 × brøkdel af højresiden
   (husk ekstra slack-variabel)
4. Find nyt optimalt tableau via dual simplex.
5. Er den fundne løsning ikke optimal, gentages proceduren, idet samtlige tilførte bibetingelser bibeholdes.

Bemærk at brøkdelen f for negative tal x er [x] + f. F.eks. –2,65 $\Rightarrow$ f = 0,35

**Rygsækproblemer – knapsack problems** (516-518)

Problemer, hvor hver variabel i kan antage værdien 0 eller 1 (valgt vs. ikke valgt), og hvor variabel i, hvis denne vælges, bidrager til z med en fast værdi $c_i$. Hvis i vælges vil den desuden besætte af $a_i$ af en given ressource.

Den grådigste algoritme:

- For hver variabel i beregnes forholdet ci/ai.
- Vælg variabel med størst $c_i/a_i$. Vælg dernæst variabel med næststørst $c_i/a_i$. Fortsæt sådan indtil den bedste tilbageværende variabel vil overskride ressourcebeholdningen. Vælg nu så meget af denne variabel som muligt.
- Lav nu branch-and-bound, idet hvert underproblem løses som i pkt. 2.

**Travelling Salesman Problems**

Problemer, hvor samtlige punkter i et netværk skal besøges med færrest mulig omkostninger.

Løsning ved branch-and-bound (522-526):

1. Opstil omkostningsmatricen
2. Løs problemet som et assignment-problem og kontrollér for delture. Findes ingen delture er den fundne løsning optimal.
3. Eksisterer delture, elimineres én af dem ved at danne nye underproblemer, hvor hver vej i delturen på skift repræsenteres ved en streg.
4. Pkt. 2 og 3 gentages via normale branch-and-bound-regler indtil en optimal løsning er fundet.

Ovenstående løsningsmodel er betinget af, at man skal vende tilbage til sit startpunket.

- Starter og slutter man i stedet uden for netværket, skal dette ekstra punkt tilføjes omkostningsmatricen (er omkostninger for dette punkt ukendt men ensartede, benyttes et passende højt tal, f.eks. 100 eller 1000).
- Skal man ikke vende tilbage til udgangspunktet og er startpunktet ligegyldigt, løses problemet som punktet ovenfor.
- Skal man ikke vende tilbage til udgangspunktet og er startpunktet fastlagt, … MANGLER…

## Kapitel 12

**Lagrange-optimering** (684-685)

Optimering af (ikke nødvendigvis lineær) objektfunktion med n variable under hensyn til m bibetingelser (af =-form).

Bestemmelse af optimal løsning:

1) Er der tale om maksimering eller minimering?
2) Opstil Lagrange-funktionen $L(x_1,\ldots,x_n,\lambda_1,\ldots,\lambda_m)$.
3) Find første-ordens-betingelser for samtlige variable (incl. $\lambda$'er).
4) Løs ligningssystemet, dvs. find $(x_1^*,\ldots,x_n^*,\lambda_1^*,\ldots,\lambda_m^*)$.
5) Er den fundne løsning optimal?
   a) Er der tale om maksimering og er objektfunktionen konkav, mens bibetingelserne er konvekse, er $(x_1^*,\ldots,x_n^*)$ optimal.
   b) Er der tale om minimering og er objektfunktionen konveks, mens bibetingelserne er konvekse, er $(x_1^*,\ldots,x_n^*)$ optimal.

**Kuhn-Tucker-optimering** (691-695)

Optimering af (ikke nødvendigvis lineær) objektfunktion med n variable under hensyn til m bibetingelser (af $\leq$ - form).

Bestemmelse af optimal løsning:

1) Omskriv evt. bibetingelser til den relevante form.
2) Er der tale om maksimering eller minimering?

3) Skal variablene ($x_1,\ldots,x_n$) være ikke-negative?
4) Opstil Kuhn-Tucker-betingelserne
   a) Svares nej i pkt. 3 benyttes teorem 9/9' s. 692.
   b) Svares ja, men vil ($x_1^*,\ldots,x_n^*$) aldrig antage negative værdier, benyttes teorem 9/9' s. 692.
   c) Svares ja, og vil ($x_1^*,\ldots,x_n^*$) kunne antage negative værdier, benyttes teorem 10/10' s. 694.
5) Løs ligningssystemet, dvs. find ($x_1^*,\ldots,x_n^*,\lambda_1^*,\ldots,\lambda_m^*$)/($x_1^*,\ldots,x_n^*,\lambda_1^*,\ldots,\lambda_m^*,\mu_1^*,\ldots,\mu_m^*$,).
6) Er den fundne løsning optimal?
   a) Er der tale om maksimering og er objektfunktionen konkav, mens bibetingelserne er konvekse, er ($x_1^*,\ldots,x_n^*$) optimal.
   b) Er der tale om minimering og er objektfunktionen konveks, mens bibetingelserne er konvekse, er ($x_1^*,\ldots,x_n^*$) optimal.

# Kapitel 16

**Deterministisk EOQ-model**

Model til beskrivelse af lagerplanlægning i en verden uden usikkerhed.

Definitioner:

$D$ = efterspørgsel pr. tidsenhed (jævnt fordelt i tid).

$K$ = setup- / ordreomkostninger.

$h$ = lageromkostninger pr. enhed pr. tidsenhed.

$p$ = pris pr.enhed.

$q$ = EOQ = ordrestørrelsen.

$L$ = lead time.

$T$ = tid mellem to ordrer.

$L_e$ = effektiv lead time.

$r$ = genbestillingspunkt (målt i lagerbeholdning).

Model med L = 0 (871 – 875):

Bestemmelse af optimal ordre-størrelse således at de totale omkostninger TC minimeres:

$$TC(q) = \frac{K \cdot D}{q} + p \cdot D + \frac{h \cdot q}{2}$$

$$r = 0$$

$$q^* = \sqrt{\frac{2 \cdot K \cdot D}{h}}$$

$$T^* = \frac{q}{D}$$

Model med L > 0 (878 – 879):

I modeller med positiv lead time er de totale omkostninger TC, den optimale ordrestørrelse q samt tiden T mellem to ordrer uændret. Nyt er:

Hvis $L \cdot D \leq q^*$:

$$r = L \cdot D$$

Hvis $L \cdot D > q^*$:

$$L_e = L - \left[\frac{L}{T^*}\right] \cdot T^*$$

$$r = L_e \cdot D$$

## Kapitel 17

### Stokastisk EOQ-model

Model til beskrivelse af lagerplanlægning i en verden med usikkerhed.

The news vendor problem – diskret efterspørgsel (901 – 904):

$c_o$ = omkostning pr. enhed ved overskudslager.

$c_u$ = omksotning pr. enhed ved underskudslager.

$D$ = stokastisk variabel for efterspørgslen.

De forventede omkostninger minimeres ved at vælge den mindste værdi $q = q^*$, der tilfredsstiller:

$$P(D \leq q^*) \geq \frac{c_u}{c_o + c_u}$$

<u>The news vendor problem – kontinuert efterspørgsel (906 –910):</u>

Svarer til problemet ovenfor – $q^*$ findes nu blot via ligningen:

$$P(D \leq q^*) = \frac{c_u}{c_o + c_u}$$

## Kapitel 20

Kapitlet angiver alternative løsninger til en række velkendte problemer, bl.a. shortest path og knapsack-problems.

Undertegnede mener ikke, at disse alternative løsningsmetoder bør foretrækkes fremfor hidtil beskrevne metoder (pensum er 20.1, 20.2, 20.4, 20.5, 20.6).

## Kapitel 22

**Simpel køteori** (1105 – 1109)

Det antages, at ankomster foregår enkeltvis og er uafhængige af hinanden, således at tiden mellem to ankomster kan beskrives ved en eksponentialfordeling med parameter λ, hvor λ angiver det forventede antal ankomster pr tidsenhed t.

Dermed følger, at antallet af ankomster indenfor en givet tidsperiode t kan beskrives ved en poissonfordeling med parameter λ.

Det er vigtigt at præcisere tidsperioden. Ønskes f.eks. analysen baseret på den dobbelte tidsperiode, dvs. $2 \cdot t$, benyttes i både eksponential- og possionfordelingen parameteren $2 \cdot \lambda$.

Eksponentialfordelingen har ingen hukommelse. Dvs. den fremtidige fordeling påvirkes ikke af, hvor lang tid, der allerede er gået.

**M / M / 1 / GD / ∞ / ∞ - køsystemet** (1124 – 1130)

Et køsystem, hvor ankomst- og servicetid er eksponentialfordelt. Der er 1 "skranke", og der tillades et uendelig antal kunder i systemet ligesom, ligesom den samlede population antages uendelig stor.

Definitioner:

$\lambda$ = parameter for ankomsttidernes eksponentialfordeling (gennemsnitlig antal ankomster pr. tidsenhed).

$\mu$ = parameter for servicetidernes eksponentialfordeling (gennemsnitlig antal services pr. tidsenhed).

$\rho$ = trafikintensitet (sandsynlighed for at skranken er optaget).

$L$ = Gennemsnitligt antal i hele systemet.

$L_q$ = Gennemsnitligt antal kunder i kø.

$L_s$ = Gennemsnitligt antal kunder, der modtager service.

$W$ = Gennemsnitligt tidsforbrug i hele systemet pr. kunde.

$W_q$ = Gennemsnitligt tidsforbrug i kø pr. kunde.

$W_s$ = Gennemsnitligt tidsforbrug i servicedelen pr. kunde.

Formler:

$$\rho = \frac{\lambda}{\mu}$$

$$L = \frac{\rho}{1-\rho} = \frac{\lambda}{\mu-\lambda} = \lambda \cdot W$$

$$L_q = \frac{\rho^2}{1-\rho} = \frac{\lambda^2}{\mu \cdot (\mu-\lambda)} = \lambda \cdot W_q$$

$$L_s = \rho = \lambda \cdot W_s$$

$$W = \frac{1}{\mu-\lambda}$$

$$W_q = \frac{\lambda}{\mu \cdot (\mu-\lambda)}$$

$$W_s = \frac{1}{\mu}$$