Erhvervsøkonomiske metoder: Dynamisk programmering

Af Ove Hedegaard*)

Dynamisk programmering forkortet DP anvendes, som navnet antyder, ved optimering af dynamiske systemer. Dvs. systemer som kræver, at der træffes en række beslutninger, og hvor de allerede trufne beslutninger indvirker på de efterfølgende.

Som eksempel på et dynamisk system kan man tage et rederi, der skal planlægge fragtruter for sine skibe. Har man først besluttet at lade et skib gå fra Rotterdam til Trinidad, vil det helt klart få betydning for skibets efterfølgende ture.

Et andet eksempel er et firma, der har begrænset kapital til rådighed, og som kan vælge mellem forskellige investeringsalternativer. En sådan situation kan betragtes som et dynamisk system, idet en beslutning om at investere en del af kapitalen i et af alternativerne muligvis medfører, at andre projekter nu må lades helt ude af betragtning, fordi der ikke længere findes tilstrækkelig kapital.

Som et tredie eksempel kan man tage et transportfirmas politik med hensyn til udskiftning af biler. De beslutninger der træffes om udskiftninger det ene år, vil helt klart påvirke beslutninger om udskiftninger de følgende år.

*) Metodeforskningsgruppen, Handelshøjskolen i København, december 1973.

Erhvervsøkonomiske metoder: Dynamisk programmering

Af Ove Hedegaard*)

Dynamisk programmering forkortet DP anvendes, som navnet antyder, ved optimering af dynamiske systemer. Dvs. systemer som kræver, at der træffes en række beslutninger, og hvor de allerede trufne beslutninger indvirker på de efterfølgende.

Som eksempel på et dynamisk system kan man tage et rederi, der skal planlægge fragtruter for sine skibe. Har man først besluttet at lade et skib gå fra Rotterdam til Trinidad, vil det helt klart få betydning for skibets efterfølgende ture.

Et andet eksempel er et firma, der har begrænset kapital til rådighed, og som kan vælge mellem forskellige investeringsalternativer. En sådan situation kan betragtes som et dynamisk system, idet en beslutning om at investere en del af kapitalen i et af alternativerne muligvis medfører, at andre projekter nu må lades helt ude af betragtning, fordi der ikke længere findes tilstrækkelig kapital.

Som et tredie eksempel kan man tage et transportfirmas politik med hensyn til udskiftning af biler. De beslutninger der træffes om udskiftninger det ene år, vil helt klart påvirke beslutninger om udskiftninger de følgende år.

*) Metodeforskningsgruppen, Handelshøjskolen i København, december 1973.

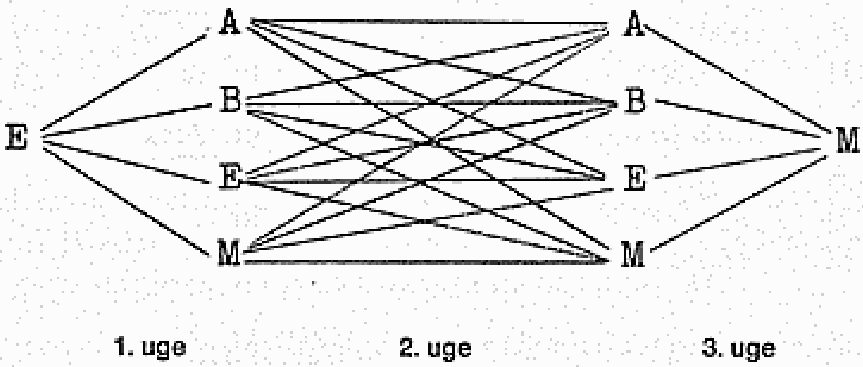
Principperne for anvendelse af dynamisk programmering er ret enkle: Man begynder simpelthen bagfra og siger, at hvis de tidligere beslutninger, optimale eller ej, har ført frem til en vis tilstand, hvad er så den optimale beslutning i denne tilstand?

Et eksempel vil sikkert gøre forståelsen lettere.

En skipper ligger med sin båd i Bergen og skal planlægge de næste ture. Han har lovet familien at være hjemme i Marstal om tre uger og skal nu forsøge at tjene så meget som muligt inden da. Han ved, at han kan få last til og fra en hvilken som helst af følgende byer: Bergen, Marstal, Edinburg og Amsterdam. Han ved også, at hver rejse tager én uge. Fortjenesten på de forskellige ruter fremgår af følgende tabel:

Fra \ Til	A	B	E	M
A	0	400	600	800
B	400	0	800	600
E	400	600	0	200
M	800	400	200	0

For at få lidt bedre overblik, kan man tegne en figur, der viser de mulige ruter skipperen kan vælge:



Anvendes dynamisk programmering på problemet, skal man angribe det bagfra. Ser man på systemet, når der er en uge tilbage af turen, er der ikke noget valg, kursen skal sættes mod Marstal, som han jo har lovet.

Derimod er det ikke klart, hvor han befinder sig efter at have sejlet rundt i to uger, og det er derfor nødvendigt, at opstille en tabel, som viser den største fortjeneste, der kan opstå i den *sidste uge* afhængig af, hvilken havn de tidligere beslutninger har bragt skibet frem til.

3. uge

Fra	Til	Fortjeneste
A	M	800,– kr.
B	M	600,– kr.
E	M	200,– kr.
M	M	0,– kr.

Dvs. ligegyldigt hvilken havn skipperen befinder sig i efter to ugers sejlads, ved man hvormeget han kan tjene den sidste uge.

Går man nu yderligere et trin tilbage i tiden, kan han i den anden uge sejle fra en hvilken som helst havn til enhvilken som helst af de fire havne. Problemet bliver nu, med udgangspunkt i hver af de fire havne, at finde den tur for *næstsidste uge*, som sammen med den deraf følgende tur for sidste uge, giver den største fortjeneste.

2. uge

Fra	Til	Fortjeneste	+ 3. uge	= Samlet fortjeneste
A	A	0 kr.	800 kr.	800 kr.
A	B	400 kr.	600 kr.	1000 kr. Max
A	E	600 kr.	200 kr.	800 kr.
A	M	800 kr.	0 kr.	800 kr.
B	A	400 kr.	800 kr.	1200 kr. Max
B	B	0 kr.	600 kr.	600 kr.
B	E	800 kr.	200 kr.	1000 kr.
B	M	600 kr.	0 kr.	600 kr.
E	A	400 kr.	800 kr.	1200 kr. Max
E	B	600 kr.	600 kr.	1200 kr. Max
E	E	0 kr.	200 kr.	200 kr.
E	M	200 kr.	0 kr.	200 kr.
M	A	800 kr.	800 kr.	1600 kr. Max
M	B	400 kr.	600 kr.	1000 kr.
M	E	200 kr.	200 kr.	400 kr.
M	M	0 kr.	0 kr.	0 kr.

Dvs. hvis skipperen efter en uge på søen ligger i Amsterdam, skal han sejle via Bergen til Marstal for at tjene mest muligt. På samme måde fås, hvis han er blevet i Bergen den første uge, at han skal sejle via Amsterdam hjem, for at få mest muligt ud af detosidste uger.

Derimod kan han, såfremt han er kommet til Edinburg, frit vælge mellem at gå via Amsterdam eller via Bergen til Marstal. Fortjenesten for de to sidste uger bliver i begge tilfælde 1200,– kr. Endelig ses det, at fra Marstal skal turen gå til Amsterdam og hjem igen.

Der mangler nu kun et trin i beslutningskæden for at den samlede fortjeneste, for alle tre uger, skal blive størst mulig. Ved begyndelsen af første uge er udgangspositionen givet, nemlig Bergen, og spørgsmålet bliver nu, om skipperen skal blive liggende, eller hvilken by han skal sejle til.

1. uge

Fra	Til	Fortjeneste	+	To sidste uger	=	Total fortjeneste
B	A	400 kr.		1000 kr.		1400 kr.
B	B	0 kr.		1200 kr.		1200 kr.
B	E	800 kr.		1200 kr.		2000 kr.
B	M	600 kr.		1600 kr.		2200 kr. Max

Hermed er den optimale plan for de tre uger klar. Det blev altså fra Bergen til Marstal videre til Amsterdam og derpå tilbage til Marstal igen, hvilket giver skipperen en samlet fortjeneste på 2200,– kr.

Havde man ikke anvendt dynamisk programmering på problemet, men optimeret på hver uge for sig, havde skipperen først sejlet til Edinburg og derved tjent 800,– kr. den første uge. Fra Edinburg ville han tage tilbage til Bergen den anden uge og tjene 600,– kr., og endelig den sidste uge ville turen gå fra Bergen til Marstal med en fortjeneste på 600,– kr., hvilket ialt giver 2000,– kr. – eller 200,– kr. mindre end den optimale rute.

De her viste principper blev første gang anvendt af den amerikanske professor R. Bellman, som opstillede en matematisk teori for »fler-trins-beslutningsprocesser«, engelsk »multistage decisions«, der kan karakteriseres ved:

1. En funktion, der viser fortjenesten ved i en vis situation at træffe en vis beslutning.
2. En eller flere mulige beslutninger.
3. Systemets nuværende tilstand.
4. Antal beslutninger som endnu skal træffes.

For denne type af problemer opstillede Bellman et optimaliseringsprincip, der siger: »Den fremtidige fortjeneste afhænger kun af, hvilken situation man er i for øjeblikket og ikke af hvordan man er kommet der.« Eller mere præcist udtrykt: En optimal politik har den egenskab, at uanset begyndelsestilstanden og de i foregående perioder trufne beslutninger, må alle de resterende dispositioner udgøre en optimal politik med hensyn til den tilstand, der er en følge af de første beslutninger.

For at danne et matematisk udtryk for Bellmans optimaliserings princip, indføres følgende betegnelser:

$f_N(x)$ = den totale gevinst af en N-trins proces, der begynder i tilstanden x, når en optimal politik følges.

P = mængden af mulige beslutninger.

$R_N(x, p)$ = fortjenesten ved i tilstand x at træffe beslutningen $p \epsilon P$ (fortjenesten kan være en funktion af N, antallet af trin der er tilbage).

$x'(N, x, p)$ = den nye tilstand som følger af at træffe beslutning p i tilstand x.

Med hjælp af de her definerede betegnelser kan optimaliseringsprincippet omdannes til følgende ligning:

$$f_N(x) = \max_{p \epsilon P}(R_N(x, p) + f_{N-1}(x'(N, x, p)))$$

som siger, at den totale fortjeneste af en N-trins proces er første trins fortjeneste, plus den optimale gevinst fra N – 1 trins processen, hvor beslutningen $p \epsilon P$ vælges således, at summen maksimeres.

Det lyder måske noget abstrakt, men er egentlig ret enkel. Man starter med det N'te og sidste trin i beslutningskæden (sidste uge af skipperens tre ugers tur). Den optimale politik afhænger kun af tilstanden x (havnen som båden ligger i efter to uger), som er resultatet af de tidligere beslutninger. Ligningen reduceres derfor til:

$$f_1(x) = \max_{p \epsilon P} R_1(x, p)$$

og man skal altså finde den beslutning p, som maksimere fortjenesten $f_1(x)$. Når beregningerne påbegyndes, kendes tilstanden x naturligvis ikke. Skipperen kunne jo ligge i en hvilken som helst af de fire havne, når der var en uge tilbage af turen. Det er derfor nødvendigt at beregne $f_1(x)$ for alle mulige værdier af x (se tabel for 3. uge). Det er en enkel opgave, da der for hver x kun findes én mulig beslutning – nemlig hjem

Går man et trin længere tilbage til næstsidste uge, lyder ligningen:

$$f_2(x) = \max R_2(x, p) + f_1(x'(2, p))$$

Der er her igen fire forskellige muligheder for x, idet skipperen jo allerede efter en uge kan befinde sig i hvilken som helst af de fire havne. Desuden har han med udgangspunkt i hver havn fire forskellige beslutninger at vælge imellem. Han kan enten blive liggende eller gå til en af de tre andre havne. Det giver ialt 16 muligheder for $R_2(x, p)$. Det sidste led $f_1(x'(2, p))$ – ja det er simpelthen fortjenesten i den sidste uge, som jo afhænger af, hvilken havn, x', han er kommet til efter 2 uger, hvilket igen afhænger af den beslutning p, som skipperen træffer, når der er to uger tilbage af turen.

Beregnes nu, for hver af de mulige tilstande, værdien af hver af de fire mulige beslutninger, fås tabellen for 2. uge. På samme måde fås tabellen for uge 1, som giver den endelige løsning på problemet.