

Heuristiske løsningsmetoder - deres relation til, og anvendelse i O. R.

Af Per Villum Hansen*)

I nærværende artikel skal de heuristiske metoders egenart og natur diskuteres. Gennem en beskrivelse af nogle af disse metoder, og ved at give eksempler på anvendelser af disse metoder, vises det - når talen går om anvendelse af disse metoder - at den enkelte metode i sig selv er ret uinteressant, men at det centrale er opbygningen af den sekvens man anvender disse metoder i.

1. Introduktion

Operationsanalyse har nu efterhånden så mange år på bagen, at den som disciplin snart kan fejre 30 års jubilæum.

Disse tredive år har heldigvis ikke undgået at sætte deres spor. Mere præcist skulle man vel hellere sige, at operationsanalysen har undergået en væsentlig udvikling igennem disse år.

Det helt markante træk, og det, der for en stor del har betinget denne udvikling, er datamaten, der explicit er draget ind i og har sat sit præg på den operationsanalytiske modeldannelse. Uden på nogen måde at overdrive kan man slå fast, at en væsentlig del af det operationsanalytiske arbejde, der er udført indtil i dag, ikke kunne have været udført, hvis ikke vi havde haft datamaten til rådighed. Hermed er indiceret, at det er datamaten, der har gjort det muligt at arbejde med, udvikle og løse omfattende og komplekse problemer med store datamængder. Dette er en udvikling på godt og ondt. Godt fordi der virkelig eksisterer problemer med en kompleks natur, som kræver komplekse modeller, for at beskrivelsen kan være adekvat, og hvor løsningen af disse modeller er mindst lige så kompleks som selve modellen. Ondt, fordi dette hjælpeværktøj kan misbruges, thi det sker ofte, at jo mindre vi forstår af et problem, jo flere variable anvender vi for at (bort)forklare vor usikkerhed. Datamaten leder til denne fristelse, fordi den gør det muligt at få hold på mange variable simultant.

*) Amanuensis, cand. merc., Metodeforskningsgruppen, Handelshøjskolen i København. Artiklen indleveret december, 1971.

Heuristiske løsningsmetoder - deres relation til, og anvendelse i O. R.

Af Per Villum Hansen*)

I nærværende artikel skal de heuristiske metoders egenart og natur diskuteres. Gennem en beskrivelse af nogle af disse metoder, og ved at give eksempler på anvendelser af disse metoder, vises det - når talen går om anvendelse af disse metoder - at den enkelte metode i sig selv er ret uinteressant, men at det centrale er opbygningen af den sekvens man anvender disse metoder i.

1. Introduktion

Operationsanalyse har nu efterhånden så mange år på bagen, at den som disciplin snart kan fejre 30 års jubilæum.

Disse tredive år har heldigvis ikke undgået at sætte deres spor. Mere præcist skulle man vel hellere sige, at operationsanalysen har undergået en væsentlig udvikling igennem disse år.

Det helt markante træk, og det, der for en stor del har betinget denne udvikling, er datamaten, der explicit er draget ind i og har sat sit præg på den operationsanalytiske modeldannelse. Uden på nogen måde at overdrive kan man slå fast, at en væsentlig del af det operationsanalysearbejde, der er udført indtil i dag, ikke kunne have været udført, hvis ikke vi havde haft datamaten til rådighed. Hermed er indiceret, at det er datamaten, der har gjort det muligt at arbejde med, udvikle og løse omfattende og komplekse problemer med store datamængder. Dette er en udvikling på godt og ondt. Godt fordi der virkelig eksisterer problemer med en kompleks natur, som kræver komplekse modeller, for at beskrivelsen kan være adekvat, og hvor løsningen af disse modeller er mindst lige så kompleks som selve modellen. Ondt, fordi dette hjælpeværktøj kan misbruges, thi det sker ofte, at jo mindre vi forstår af et problem, jo flere variable anvender vi for at (bort)forklare vor usikkerhed. Datamaten leder til denne fristelse, fordi den gør det muligt at få hold på mange variable simultant.

*) Amanuensis, cand. merc., Metodeforskningsgruppen, Handelshøjskolen i København. Artiklen indleveret december, 1971.

Disse forhold rejser problemer af tilsyneladende forskellig karakter. Først, selv om vi arbejder med store systemer, og i dette arbejde anvender datamaten som hjælpeværktøj, er der en overgrænse for, hvor store problemerne kan være, alene på grund af den tid, det tager på datamaten. D.v.s. at der er et behov for en række beslutningsregler, der hurtigt og effektivt kan reducere dimensionen i problemet. Dernæst, på grund af den til tider betragtelige regnetid, udvikler vi nye beslutningsregler med det formål at reducere denne. Dette betyder, at vi, i det mindste implicit, har fået lagt et andet indhold i vort optimeringsbegreb, således at vi fra at optimere *modellen*, arbejder hen mod at optimere model *plus* løsning. Endelig kommer det krav, at vi som problemløsere skal kunne forstå den fulde rækkevidde af vore modelmanipulationer, også for disse komplekse problemers vedkommende.

Selv om disse problemer tilsyneladende er af forskellig karakter, har de imidlertid mindst et fælles element, nemlig at beslutningsreglerne må være enkle af natur. Skal vi hurtigt reducere dimensionen i et problem, må generalisationsgraden være stor, d.v.s. at reglen må være enkel. Skal computertiden reduceres, må antallet af „checkpoints“ i hver beregning reduceres, d.v.s. at reglen skal være enkel. Skal vi forstå den fulde rækkevidde af vore modelmanipulationer, må vore datarelationer ikke være for omfattende, d.v.s. reglen må være enkel.

Lad det med det samme være understreget, at selv om disse regler må være enkle, er dette langt fra, snarere tværtimod, ensbetydende med, at det er *let* at finde regler af denne type, der er relevante og egnede på et vilkårligt, konkret problem.

Et sæt regler, der opfylder ovennævnte krav, eller i det mindste synes at opfylde disse krav, går under betegnelsen heuristiske beslutningsregler. I det følgende skal nogle af disse reglers egenart og natur behandles.

2. Om begrebet heuristik

Begrebet heuristisk blev introduceret i operationsanalysen af Simon og Newell i 1958, (Newell, Simon, 1958).

De gav ikke dengang nogen klar definition af begrebet. Selv i dag må vi konstatere, at vi begrebsmæssigt står i samme situation, idet der ikke er nogen forfatter, der har givet en klar definition af begrebet, i det mindste ikke nogen anerkendt definition.

Ser vi på antallet af forfattere, der har *anvendt* heuristiske beslutningsregler, er dette antal betragteligt, og så er vi vel i den situation, hvor den klare definition af begrebet for alle praktiske formål er overflødig.

For dem, der har svært ved at acceptere dette synspunkt, kan det fremhæves, at der er ingen, der nogensinde har bragt en klar definition af begrebet operationsanalyse. Alligevel ved vi alle, hvad operationsanalyse er, fordi der efterhånden er så mange, der har arbejdet med operationsanalyse, så vi gennem dette arbejde har fået givet ordet mening. For alligevel at give et indtryk af begrebet heuristisk skal der bringes et udpluk af nogle opfattelser, der er på markedet.

Newel, Shaw, Simon (1958, s. 22). "We use the term heuristic to denote any principle or device that contributes to the reduction in the average search to solution."

Kuehn, Hamburger (1963, s. 644). "Heuristic programs . . . an approach to problem solving where emphasis is on working towards optimum solutions procedures, rather than the optimal solutions."

Tonge (1961, s. 231). "Heuristic programming is an art. It has advanced to date through the construction of particular programs, the examination of these programs and their behaviour at the construction of better programs based on the insight thus gained . . . at present this is a body of knowledge built up through experience with specific examples, lacking as yet an underlying analytic framework."

Tonge (1963, s. 126) siger om *resultatet* af heuristiske overvejelser: "no guaranty of a satisfactory solution or, often, of any solution."

Vazonyi (1967, s. 126). "We do not have an algorithm, so we cannot be assured that we find a solution . . . consequently a selected search technique is employed by recognizing patterns of improvements in solutions."

Det ses, at fælleselementerne i disse opfattelser er dels, at man explicit forsøger at lære af den information, der kommer på bordet under søgningen, dels at resultatet af disse overvejelser ikke nødvendigvis vil medføre en løsning af en nærmere specificeret art, f. eks. en optimal løsning.

Kontrastes algoritme og heuristik, har vi, at en algoritme garanterer en optimal løsning, i modsætning til en heuristik, medens der i en heuristik, i modsætning til en algoritme, forsøges indbygget en læremekanisme.

Desværre gælder Tonges ord stadig i dag " . . . at present this is a body of knowledge built up through experience with examples, lacking as yet an analytic framework."

Dette skyldes, så vidt nærværende forfatter kan se, at vi ikke ved nok om, hvad det vil sige „at lære“. Vi ved en masse om, hvad det ikke er, f. eks. akkumulation af facts, akkumulation af metoder, etc. Omvendt ved vi, hvornår et system *har* lært sig noget, f. eks. når det har ændret adfærd overfor en given impuls. Det, der er imellem disse to situationer, er en dunkelt oplyst blackbox.

Manglende kendskab til disse fænomener skal dog ikke afholde os fra at forsøge at trække nogle basale principper ud af rækken af eksempler på anvendelsen af heuristik. Dette at et fænomen er diffust, kan være en egenskab ved fænomenet, og behøver derfor ikke at være en egenskab ved beskrivelsen.

3. Metodekarakteristika

Før end vi går ind i en beskrivelse af de heuristiske metoder, må vi lægge os fast på et beskrivelsessæt, eller en måde at repræsentere vore metoder på.

Et beskrivelsessæt, der synes hensigtsmæssigt for vort formål, er følgende tredimensionale beskrivelsesform:

- problemrepræsentation
- løsningsprocedure
- verifikation

Vi vil definere indholdet i disse tre begreber gennem et eksempel, nemlig løsningen af en andengradsligning.

Problemrepræsentation:

find x , således at $ax^2 + bx + c = 0$

Procedure:

beregn $x = b/2a \pm 1/2a\sqrt{b^2 - 4ac}$

Verifikation:

indsæt ovenstående udtryk for x i $ax^2 + bx + c$ og vis, at resultatet bliver nul.

Newel (1969), der har foreslået denne beskrivelsesform, mener, at disse tre dimensioner er fundamentale træk ved alle metoder.

4. Nogle heuristiske metoder

De metoder, der er valgt ud for en nærmere beskrivelse, er "Generate and Test", "Match", samt "Hill Climbing".

4.1. GENERATE AND TEST

Som termen indicerer, har metoden to elementer, en generator til generering af løsningsforslag, samt et test der afgør, hvorvidt en foreslået løsning er en løsning.

I dens mest simple form kan generatoren producere løsningsforslag rent tilfældigt, en art "random walk". Det klassiske eksempel her er åbningen af en kombinationslås. Testet kan i dette tilfælde være en konstatering, om døren går op eller ej.

Efterhånden som man lærer sig mere om problemets natur, kan man gå ind og gøre genereringsreglerne mere sofistikerede, udnyttende den viden man har erhvervet, f. eks. ved kun at generere løsningsforslag i en interessant del af løsningsrummet.

Vender man sig mod problemløsende systemers løsningsadfærd, vil man ofte notere, at "generate and test" anvendes som præliminær metode, idet anvendelsen af metoden kan give systemet et indtryk af problemets natur. En arkitekt vil f. eks. som oftest starte ud med at lave alternative løse rids af den konstruktion, han arbejder på, vurdere disse, bestemme sig for en af dem, og arbejde videre på dette grundlag, lave nye udkast, etc.

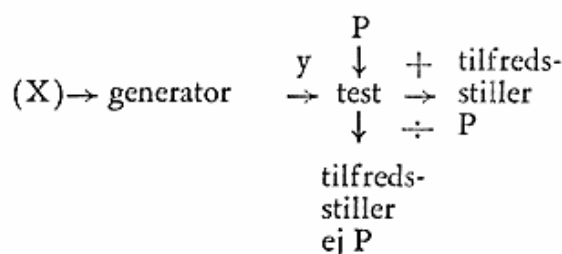
Formaliseres metoden i ovennævnte beskrivelsessæt, har vi:

4.1.1. PROBLEMREPRÆSENTATIONEN:

givet: En generator der producerer en mængde (X)
 En løsningsmængde P

find: Elementer i (X) der tilfredsstillter P(x).

4.1.2. PROCEDURE:



4.1.3. VERIFIKATION:

y er en løsning, hvis og kun hvis $y \in X$ og $P(y)$

- 1) $y/\text{løsning}$ hvis og kun hvis $\text{test} \implies y/+$
- 2) $\text{test} \implies y/+$ hvis og kun hvis $P(y)$ med input /y
- 3) input/y hvis og kun hvis $\text{generator} \rightarrow y/\text{element}$
- 4) $\text{generator} \rightarrow y/\text{element}$ hvis og kun hvis $y \in (X)$
- 5) $y \in (X)$ implicerer, at hændelsen $\text{generator} \rightarrow y/\text{element}$ vil finde sted før eller senere

Det ses, at pkt. 5 ovenfor er det kritiske punkt, thi der er ingen garanti for, at elementet y vil blive genereret inden for en overskuelig fremtid, selv om y er indeholdt i mængden x.

4.2. MATCH:

Mange gange vil det være således, at man på forhånd kan opstille nogle krav, en løsning skal opfylde for at være f. eks. en optimal, tilfredsstillende eller en mulig løsning.

Dette kan f. eks. være en række begrænsninger, der skal tilfredsstillende. Er begrænsningerne tilfredsstillende allesammen, har vi en mulig løsning.

På den anden side kan man f. eks. have en efterspørgselsfaktor, der skal tilfredsstillende, og kan således være interesseret i at undersøge, i hvilket omfang efterspørgselsfaktoren kan tilfredsstillende med de begrænsninger, der eksisterer for en produktionsafdeling. Dette er en typisk „Match“-problematik. Beskriver vi f. eks. begrænsninger ved F og efterspørgselskravene ved X , kan vi undersøge, om X er indeholdt i F , d.v.s. forsøge at „matche“ F og X .

Der er to forudsætninger, der her må være opfyldt, førend X og F kan matches. Den første er, at de måles i samme dimension. I ovenstående eksempel kan en dimension være timer. Den anden forudsætning er, at det må være muligt at generere delmængder af X og F , således at X og F er identiske, hvis og kun hvis delmængden af X og F er identiske. Denne sidste forudsætning kan i nogle tilfælde være noget problematisk, for der forudsættes additivt funktionsssammenhæng uafhængig af den konkrete opdeling af delmængder.

I den mest simple form vil metoden som ovenfor beskrevet foretage en simpel sammenligning. Det er imidlertid muligt, hvis denne sammenligning falder negativt ud, at forsøge at transformere X og F over i andre enheder og på dette grundlag foretage en ny match-procedure. For mange tilfælde vil det således være frugtbart at have en transformations vektor $\{v\}$.

Herved forøger man ikke alene sine chancer for at få en løsning, men man har samtidig mulighed for at få flere løsninger, d.v.s. finde, at X er indeholdt i F , i flere dimensioner.

Man kan åbenbart lære sig noget om X set i forhold til F , i retning af fællestræk (pattern recognition).

Metoden kan formaliseret beskrives som følger:

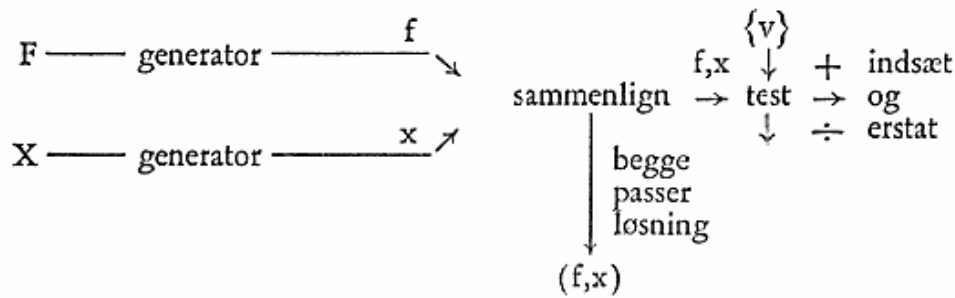
4.2.1. PROBLEMREPRÆSENTATION:

Givet: et sæt variable

en form F , som er et udtryk indeholdende variable,
et udtryk X .

Find: om X er indeholdt i et sæt, defineret af F . d.v.s. find værdier i $\{v\}$, således at $X = F$, når disse værdier indsættes.

4.2.2. PROCEDURE:



4.2.3. VERIFIKATION:

Sættet (f,x) er en løsning, hvis og kun hvis $(x) = X$ og $(f) = F$ og $(x) = (f)$.

- I 1: sættet (f,x) /løsning har som nødvendig betingelse sammenligning \rightarrow element $x =$ element f
- II 1: sættet (f,x) /løsning har som tilstrækkelig betingelse \rightarrow alle $(x) =$ alle (f)
- II 2: Sammenligning/alle $(x) =$ alle (f) hvis og kun hvis generator \rightarrow alle (x) samt generator \rightarrow alle (f)
- II 2.1. Generator \Rightarrow alle (x) , hvis og kun hvis der eksisterer et x^1 i X som \complement generator \rightarrow alle (x) hvor $x^1 = \emptyset$. Generator \rightarrow alle (f) , hvis og kun hvis der eksisterer et f^1 i F som \complement generator \rightarrow alle (f) , hvor $f^1 = \emptyset$
- I 2: sammenligning \rightarrow element $x =$ element f , hvis og kun hvis test $\rightarrow x(v) = f(v)/+$
- I 3: test $\rightarrow x(v) = f(v)/+$, hvis og kun
 - hvis $\sqsupset \{x,v\} \rightarrow x(v)$
 - samt $\sqsupset \{f,v\} \rightarrow f(v)$
- I 4: $\sqsupset \{x,v\} \rightarrow x(v)$ samt $\sqsupset (f,v) \rightarrow f(v)$ implicerer, at hvis x og f er målelig i v , vil hændelsen test $\rightarrow x(v) = f(v)/+$ før eller senere finde sted.

Pkt. I.4 er det kritiske punkt, idet det ses, at selv om både x og f er målelige i v , kan genereringen af x -elementet være en delmængde af x , der ikke matcher med delmængder f af F f. eks. fordi $x(v) > f(v)$ eller $x(v) < f(v)$. D.v.s. afhængig af, hvorledes vi skærer i X og F , hvilke delmængder vi får frem, kan vi få en løsning, subs. ikke en løsning.

4.3. HILL-CLIMBING:

Hill-climbing er som navnet indikerer en metode til at finde store værdier i et løsningsrum, f. eks. den optimale løsning. I princippet virker den som generate and testprocedure, men med den modifikation, at den har indbygget en hukommelse, således at den lagrer hele tiden bedst fundne element.

Man kan sige, at det er en af de mest elementære optimeringsmetoder. Det enkelte genererede element tilforordnes to egenskaber, nemlig variabelværdien x og en funktionsværdi $f(x)$. Der er i metoden et sæt generatorer (g), der med udgangspunkt i den eksisterende løsning bestemmer, hvor det nye punkt, der skal sammenlignes med, skal findes. Dette sæt af generatorer kan designes på mange måder. F. eks. kan det være en tilfældighedsgenerator, det kan være en gradientgenerator, det kan være en "step length" generator, etc. Der skal ikke her gås ind i en nærmere beskrivelse af disse ting.

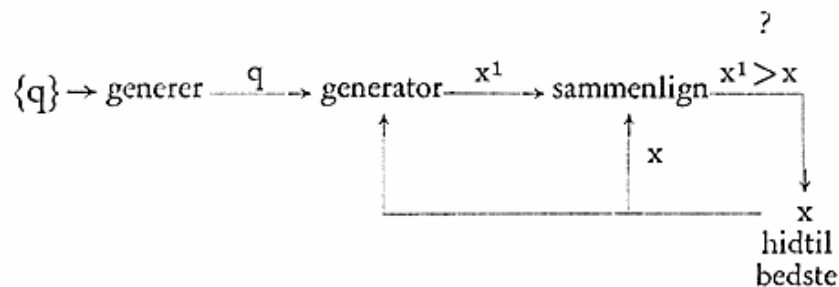
Formaliseres metoden, fås følgende:

4.3.1. PROBLEMREPRÆSENTATION

Givet: en sammenligning af de to elementer i et sæt $\{x\}$ for at bestemme, hvilket element der er størst. Et sæt generatorer $\{q\}$ hvor $q(x) = x^1$ givet et allerede fundet element x .

Find: det største $x \in \{x\}$

4.3.2. PROCEDURE



4.3.3. VERIFIKATION:

$x \in X$ er den bedste løsning, hvis og kun hvis $\nexists x^1 > x$ i $\{X\}$, hvor $x^1 = \Phi$

- 1: bedste løsning/ x hvis og kun hvis sammenlign./ $X \rightarrow x^1 < x$
- 2: sammenlign./ $X \rightarrow x^1 < x$, hvis og kun hvis operator $q(x) = x$
- 3: operator $q(x) = x$, hvis og kun hvis generator $\rightarrow q/q(x)$
- 4: generator $\rightarrow q/q(x) = x$, hvis og kun hvis $q/q(x) = x \in \{q\}$
- 5: $q/q(x) = x \in \{q\}$ implicerer, at før eller senere vil der blive genereret en operator q , hvor $q(x) = x$.

Det kritiske punkt her bliver pkt. 5, idet det, selv om der før eller senere vil blive genereret en generator, og hvor $q(x)$ før eller senere vil producere det omtalte x , ikke er givet, at disse hændelser begge vil ske indenfor en realistisk tid.

Ovennævnte metodegennemgang bygger især på Newel (1969) og Wheeling (1969). For yderligere detaljer se disse referencer.

5. Diskussion af de omtalte metoder

Som det vil fremgå af ovennævnte gennemgang, er hver enkelt metode alene meget elementær og enkel, så enkel og elementær at hvis den står alene i ovennævnte form, vil den have meget begrænsede anvendelsesmuligheder til alle praktiske formål.

Ideen er imidlertid, at de generatorer og operatorer, der er nævnt, kan tilpasses det konkrete problem på en sådan måde, at anvendelsesmulighederne bliver realistiske.

Derudover vil det i praktiske situationer være således, at ikke blot en af disse metoder, men et hierarki i forskellige udformninger anvendes. Hierarkiet bliver designet på en sådan måde, at søgeprocessen bliver mere og mere centreret omkring det punkt, hvor den optimale løsning ligger, d.v.s. at man har en stedse konvergens mod den optimale løsning, men stadig ingen garanti for at nå den.

Selv om generatorer og operatorer designes specielt til det foreliggende problem, og selv om der skabes et hierarki af disse metoder, ændrer dette ikke på den enkelte metodes struktur, den bevarer stadig sin enkelhed. D.v.s. at vi ved en sådan metodeanvendelse får honoreret alle de krav om enkelhed, vi opstillede i begyndelsen af denne artikel.

6. Heuristisk løsning af et produktionsplanlægningsproblem

Til illustration af, og som en slags dokumentation for ovenstående, skal et konkret eksempel på anvendelse af heuristiske løsningsmetoder bringes og diskuteres.

Det produktionsplanlægningsproblem, der er refereret til, er beskrevet i *Villum Hansen* (1969) og er i korthed følgende: Under forudsætning af en kendt deterministisk efterspørgsel for produktet i ($i = 1, 2, \dots, p$), der må tilgodeses, ønskes produceret en produktionsplan, hvori der tages hensyn til, at det enkelte produkt skal behandles på j maskiner ($j = 1, 2, \dots, n$), hver med en kendt og given kapacitet. Interessehorisonten er t ($t = 1, 2, \dots, T$) perioder.

Lidt mere formaliseret kan problemet udtrykkes

$$\min: \sum_t \sum_i \frac{X_{it} \cdot O_i \cdot R}{X_{it} \cdot 2 \cdot T} + \sum_t \sum_i (y_{it} + d_{it}) O_i \cdot R_{/T}$$

$$\text{u.h.t. } \sum_i (X_{it} \cdot b_{ij} + Y_{it} \cdot a_{ij}) \leq c_{jt} : j = 1, 2, \dots, m; t = 1, 2, \dots, T$$

$$\sum_t Y_{it} \geq \sum_t d_{it} \quad i = 1, 2, \dots, p$$

hvor Y_{it} = antal producerede enheder af produkt i , i delperioden.

X_{it} = antal set-up for produkt i , i delperiode t

O_i = variable omkostning for produkt i

d_{it} = efterspørgsel for produkt i , i delperiode t

b_{ij} = set-up tid for produkt i på maskine j

a_{ij} = produktionstid for produkt i på maskine k

c_j = kapacitet til rådighed på maskine j i delperiode t

R = rentefaktor

T = antallet af delperioder

m = antallet af maskiner

p = antallet af produkter

Kriteriefunktionens første led udtrykker, at lageromkostningerne, som er en funktion af antallet af set-up, skal minimeres. Det andet led udtrykker, at givet at hvis der ikke er kapacitet i en delperiode til dækning af efterspørgslen, skal det produkt, der er billigst, målt i lageromkostninger, flyttes tilbage i tid, for produktion der.

Den første restriktion er en kapacitetsrestriktion.

Den anden restriktion fortæller, at efterspørgslen skal honoreres.

For en nærmere diskussion af denne formulering se *Villum Hansen* (1969 s. 47-55).

Til løsning af det formulerede problem blev der udviklet følgende hierarki af heuristikker.

- I
- (i) $CPK_{jt} = \sum_j b_{ij} + a_{ij} \cdot d_{it} \quad i = 1, 2, \dots, p$
 - (ii) $CRK_{jt} = CR_j - CPK_{jt}$

Det ses, at der her er anvendt en matchprocedure. Vi kender den kapacitet, der er til rådighed CR_j (svarer til F i afsnittet om matchprocedure), og vi ønsker at undersøge, om en del-periodes efterspørgsel kan dækkes af samme delperiodes produktion. D.v.s. at vi må først finde det kapacitetskrav CPK_{jt} - dette kræver (svarer til x i afsnittet om matchproceduren), og derefter matche de to størrelser CRK_{jt} og CR_j . Testet er her simpel subtraktion, og værdien er tidsenheder. I generatorerne anvendes teknikken explicit enumeration. Match-resultatet gemmes i CRK-Matrisen, d.v.s. den kapacitet, der er tilovers efter matchproceduren målt med fortegn.

$$\begin{aligned} \text{II} \quad & CRK_{jt} \rightarrow > 0 \rightarrow \text{III} \\ & \quad \quad \quad \swarrow < 0 \rightarrow \left\{ \begin{array}{l} \text{---} (j) \left\{ \min, (o_i/a_{ij} \mid a_{ij} > 0, d_{it} > 0) \right\} \rightarrow \\ \rightarrow \left\{ \text{---} (d_{it}) \rightarrow d_{it} \rightarrow d_{it} \div CRK_{jt/a_{ij}} \right\} \end{array} \right\} \end{aligned}$$

På matchresultatet prøves en generate and test procedure, for at afsløre om der er negative elementer. Er der ikke det, fortsættes til pkt. III. Er der negative elementer, applikeres hill-climbing metoden for at finde det produkt, hvis produktion skal flyttes, for at elementerne i CRK_{jt} kan blive ikke-negative. Operatoren q , der bestemmer, hvilke x der skal genereres, ses at være bestemt af en kombineret match og generate and test procedure, d.v.s. hvorvidt d_{ix} og a_{ij} hver for sig er større end nul, samt om der kan findes et i , hvor de begge er større end nul.

Testet, der afgør, hvilket i der skal vælges (det i , der minimerer andet led i kriteriefunktionen), bestemmes af forholdet o_i/a_{ij} . Udfaldet af disse test medfører justeringen

$$\rightarrow \left\{ d_{it} \right\} \rightarrow d_{it} \rightarrow d_{it} \div CRK_{jt/a_{ij}}$$

$$\begin{aligned} \text{III} \quad & \max_i \left\{ \left((o_{it} \cdot d_{it}/X_{it}) \div (o_{it} \cdot d_{it}/X_{it} \div 1) \right) / b_{ij} \mid b_{ij} > 0 \right\} \\ & \rightarrow \left\{ CRK_j \rightarrow CRK_j \div b_{ij} \right\} \end{aligned}$$

Var der ikke negative elementer i CRK_j , kunne optimeringsprocessen af kriteriefunktionens første led starte. Det ses, at der er en hill-climbingprocedure med testfunktionen

$$\left((o_{it} \cdot d_{it}/X_{it}) \div (o_{it} \cdot d_{it}/X_{it} \div 1) \right)$$

Operatoren findes gennem en generate and test procedure ($b_{ij} > 0$). Udfaldet af disse tests medfører justeringen

$$\left\{ \rightarrow CRK_j \rightarrow CRK_j \div b_{ij} \right\}$$

Udover disse heuristikker, der er nævnt ovenfor, kommer en række generate and test samt matchprocedurer, der sikrer, at CRK_j 's elementer

vil blive nul, under hensyn til de nævnte hill climbingprocedurer. Når CRK_j 's elementer er nul, skal vi sikre os, at vi har en næroptimal løsning. Dette gøres ved at beregne skyggepriser for de enkelte delperioder SHP_t .

$$\begin{aligned}
 \text{IV} \quad & \{ \exists (t) \mid SHP_t \div SHP_{t-1} < \div a \} \rightarrow \\
 & \{ \exists (d_{it}) \rightarrow (\min_i (0_i/a_{ij} \mid a_{ij} > 0 \cap CRK_{j,t-1} \geq d_{it} \cdot a_{ij}/S) \rightarrow \\
 & \rightarrow \{ SHP_t^1, SHP_{t-1}^1 \} \rightarrow \{ ((SHP_t \div SHP_{t-1}) - \\
 & (SHP_{t-1} - SHP_{t-1}^1)) \geq a \} \rightarrow \{ d_{it} \rightarrow d_{it} - d_{it}/S \}
 \end{aligned}$$

Er forskellen i disse skyggepriser stor, er det ikke en næroptimal løsning, hvorfor vi må forsøge at nivellere skyggepriserne. Dette gøres ved at flytte produktion fra de perioder med høj skyggepris til de perioder med lav skyggepris. Proceduren, der anvendes, er en hill-climbing-procedure med testfunktionen $0_i/a_{ij}$. Resultat af denne hill-climbing-procedure testes ved at generere nye skyggepriser SHP^1 og sammenligne disse med de gamle. Opnås en forbedring, implementeres denne i den foreløbige løsning, de heraf nødvendiggjorte ændringer noteres, og der springes tilbage i hierarkiet af heuristikker, og optimeringen fastsættes fra dette nye punkt.

Før eller senere vil forskellene i skyggepriserne være så små, at det for alle praktiske formål er en „optimal“ løsning.

Ovenstående heuristikker er programmeret, og der er foretaget en testkørsel for $i = 29$ $j = 13$ og $t = 4$. Kørselstiden var 0.64 min.

Resultatet af kørslen af en algoritme udviklet af Djieninsky, Baker og Manne for $i = 35$ $j = 2$ og $t = 3$, var på 3.85 min. D.v.s. at heuristikløsningen er betydelig hurtigere, idet det er antallet af maskiner (j) og antallet af delperioder (t), der er bestemmende for kørselstiden.

Med hensyn til løsningen giver heuristikløsningen kun garanti for en konvergens med den optimale løsning, medens den ovenfor omtalte algoritme sikrer den optimale løsning.

Man må så afveje kort kørsel/næroptimal løsning med lang kørsel/optimal løsning.

7. Konklusion

Ovenstående eksempel viser, at det er muligt selv ved anvendelsen af meget simple metoder at opnå fornuftige og hurtige resultater.

Fordelene ved anvendelsen af sådanne metoder er bl. a. dem, der er fremhævet tidligere. Dette synes ikke at være uvæsentlige fordele.

Udover de heuristikker, der er nævnt ovenfor, findes en række andre. F. eks. kan nævnes mønstergenkendende heuristikker, heuristisk søg-

ning, induktive heuristikker, lærende heuristikker. Fælles for disse er, at de hver for sig kun har været anvendt et begrænset antal gange på meget simple problemer. Det har således ikke været muligt at finde træk i disse metoder, der kan betitles basale kendetegn. Før eller senere vil erfaringsmaterialet være så stort, at det bliver muligt, således at man vil komme i den situation, at mange nikker samstemmende med Newell ihukommende hans ord i 1958 "Heuristic problem Solving: the next advance in operations research".

8. Referencer

- Dzielinsky, Baker and Manne,
"Simulation test of lot-size programming",
Management Science, V. 9, nr. 2, s. 229-59, 1963.
- Kuehn, A. and Hamburger, M.,
"A heuristic program for locating warehouse",
Management Science, V. 9, nr. 4, s. 643-66, 1963.
- Newell, A., Shaw, J. and Simon H. A.,
The processes of creative thinking,
Rand Corporation, P-1320, 1959.
- Newell, A.,
"Heuristic Programming: Ill-structured Problems" i Aronofsky, J.,
Progress in Operations Research, John Wiley & Sons, N.Y. 1969.
- Tonge, F.,
"The use of Heuristic Programming in Management Science",
Management Science, V. 7, nr. 3, s. 231-37, 1961.
- Tonge, F.,
"A Heuristic Line balancing Procedure" i Fiengenbaum and Feldmann,
Computers and Thoughts, McGraw-Hill, N.Y. 1963.
- Vazoyni, A.,
"Computer aided planning, command and control" i Fisk.(ed),
The psychology of management decisions, Studentlitteratur, Lund 1967.
- Simon, H. A. and Newell, A.,
"Heuristic Problem Solving: The next advance in Operations Research",
Operations Research, V. 6, nr. 1, s. 1-10, 1958.
- Wheeling, R. F.,
"Heuristic search: Structured Problems" i Aronofsky, *Progress in Operations Research*, John Wiley & Sons, N.Y. 1969.
- Hansen, Villum,
Et produktionsplanlægningsproblem,
Handelshøjskolen, Kbh. 1969.

ning, induktive heuristikker, lærende heuristikker. Fælles for disse er, at de hver for sig kun har været anvendt et begrænset antal gange på meget simple problemer. Det har således ikke været muligt at finde træk i disse metoder, der kan betitles basale kendetegn. Før eller senere vil erfaringsmaterialet være så stort, at det bliver muligt, således at man vil komme i den situation, at mange nikker samstemmende med Newell ihukommende hans ord i 1958 "Heuristic problem Solving: the next advance in operations research".

8. Referencer

- Dzielinsky, Baker and Manne,
"Simulation test of lot-size programming",
Management Science, V. 9, nr. 2, s. 229-59, 1963.
- Kuehn, A. and Hamburger, M.,
"A heuristic program for locating warehouse",
Management Science, V. 9, nr. 4, s. 643-66, 1963.
- Newell, A., Shaw, J. and Simon H. A.,
The processes of creative thinking,
Rand Corporation, P-1320, 1959.
- Newell, A.,
"Heuristic Programming: Ill-structured Problems" i Aronofsky, J.,
Progress in Operations Research, John Wiley & Sons, N.Y. 1969.
- Tonge, F.,
"The use of Heuristic Programming in Management Science",
Management Science, V. 7, nr. 3, s. 231-37, 1961.
- Tonge, F.,
"A Heuristic Line balancing Procedure" i Fiengenbaum and Feldmann,
Computers and Thoughts, McGraw-Hill, N.Y. 1963.
- Vazoyni, A.,
"Computer aided planning, command and control" i Fisk.(ed),
The psychology of management decisions, Studentlitteratur, Lund 1967.
- Simon, H. A. and Newell, A.,
"Heuristic Problem Solving: The next advance in Operations Research",
Operations Research, V. 6, nr. 1, s. 1-10, 1958.
- Wheeling, R. F.,
"Heuristic search: Structured Problems" i Aronofsky, *Progress in Operations Research*, John Wiley & Sons, N.Y. 1969.
- Hansen, Villum,
Et produktionsplanlægningsproblem,
Handelshøjskolen, Kbh. 1969.