



Tak for den positive modtagelse

ORbit

medlemsblad for
Dansk Selskab
for Operationsanalyse

Redaktion:

Ansv. Rene Munk Jørgensen (rmj)
Jesper Larsen (jla)

IMM, bygn. 305 - DTU
DK-2800 Kgs. Lyngby
Telefon: +45 4525 3385
Fax: +45 4588 2673
E-mail: orbit@dorsnet.dk

Næste deadline:

1. maj 2003

Tryk:

Paritas Digital Service ApS
Oplag: 300
ISSN 1601-8893

Teknisk assistance:

Britt Morelli Hansen

Annoncepriser

Pr. 1/4 side: 500,- kr.
Pr. 1/2 side: 1000,- kr.
Pr. 1/1 side: 2000,- kr.
Medlemmer af DORS får 50% rabat.

Forside illustration:

Dansk Leopard 2 kampvogn. Foto
venligst stillet til rådighed af Forsvar-
skommandoen.

En af DORS' opgaver er at fremme kendskabet til operationsanalysen. For en lille forening og et relativt lille forskningsområde er dette selvsagt ikke let.

Igennem DORS' historie har man gentagne gange troet at nu var operationsanalysens gennembrud lige rundt om hjørnet. Dansk OR's mest fremtrædende folk har kæmpet indædt for derigennem at stå stærkt når OR's endelige gennembrud ville vise sig.

Skuffelsen har været stor for gennembrudet er aldrig rigtig kommet – slet ikke i Danmark. Selskabet har endog en overgang været ved at nedlægge sig selv. Sådan gik det heldigvis ikke.

Igennem de senere år har flere forhold atter sået sæden til optimisme påny:

- Computerkraft fåes nu i rigelige mængder til næsten ingen penge.
- Nye landvindinger indenfor såvel eksaktere metoder som heuristikker har givet operationsanalytikerne nye værktøjer og forbedret gamle.
- Data er digital – klar til behandling.

Det er således ingen tilfældighed at Carmen Systems i Sverige er blevet kåret til året IT-virksomhed i 2003.

Aktuelt om DORS februar 2003

Medlemsskab

Kontingentsatser indtil 31/12 2003

Personlige medlemmer
(incl. ph.d.-studerende): 250 kr./år
Studerende: 50 kr./år
Firmamedlemmer: 3000 kr./år
Institutmedlemmer: 1500 kr./år

Indbetales på Giro 9123865 (reg.nr. 1199)

Sekretariat

DORS
IMM, Bygning 305
Danmarks Tekniske Universitet,
2800 Kgs. Lyngby
e-mail: dors@dorsnet.dk

Internet

<http://www.dorsnet.dk>

Formanden har ordet ...

Det er tankevækkende, at ORbit allerede her i tredje nummer gør op med sit hidtidige layout. Det vidner om et stort engagement fra redaktørernes side, og selv om vi var mange, der var begejstrede over opsætningen i de to første numre, så er det nye layout i en klasse for sig selv takket være Britt Morelli Hansens indsats.

At 'gøre tingene rigtigt' er et legitimt mål for veldefinerede processer. Men overordnet set er målet for virksomheder og institutioner at 'gøre rigtige ting'. Bl.a. dette forhold behandles i Professor Börje Langefors meget interessante artikel om målstyring inde i bladet.

Omkring det at 'gøre tingene rigtigt' og 'gøre rigtige ting' er det min erfaring, at anvendelsen af operationsanalytiske metoder åbner for en gennemsigtig og fleksibel planlægning og styring, som netop gør det muligt at 'gøre rigtige ting'. Operationsanalysen løser således ikke blot erkendte problemer - den åbner ofte for helt nye strategiske muligheder.

Ligesom sidst er der i dette nummer to spændende artikler, som omhandler praktisk anvendelse af operationsanalytiske metoder. Som det fremgår af artiklerne er den matematiske modellering central. Medarbejderne i virksomheden leverer helt afgørende input til modellen, uden hvilket resultaterne ville være ubrugelige. Arbejdet med modellen gør på den anden side op med vanetænkning og udfordrer medarbejdere og nuværende processer på konstruktiv vis.

Siden sidste nummer af ORbit har DORS afholdt en række spændende arrangementer. Der har været en god interesse - men de kan sagtens rumme flere deltagere. Så her til sidst en opfordring til at melde sig til arrangementerne - og tag gerne en eller flere kollegaer med. Det er helt legalt at komme, selvom man endnu ikke er medlem.

Søren K. Nielsen
Formand for DORS

Indhold

Leder	2
Formanden har ordet ...	3
Militære operationsanalytikere optimerer tjenesteplaner til S-tog	4
Ny bog om ruteplanlægning	7
DORS' 40 års jubilæum fejret i november	9
Kommentar	10
Produktionsplanlægning hos Tulip med SAS/OR	14
Constraint Programming versus Mathematical Programming	16
Brian Kallehauge vinder DORS-prisen	24

Medlem af flg. internationale organisationer

- EURO: The Association of European Operational Research Societies
- IFORS: International Federation of Operations Research Societies

Institutmedlemmer

- Afdeling for Operationsanalyse, KU
- Afdelingen for Operationsanalyse, AU
- Center for Trafik og Transport, DTU
- Datalogisk Institut, KU
- Informatik og Matematisk Modellering, DTU
- Institut for Driftsøkonomi og Logistik, Handelshøjskolen i Århus
- Institut for Organisation og Ledelse, SDU

Firmamedlemmer

- Carl Bro a/s, IT og Telekommunikation
- Carmen Consulting Denmark
- COWI
- Elkraft a.m.b.a.
- eGruppen
- Forsvarets Forskningstjeneste
- MOSEK
- Transvision A/S

Militære operationsanalytikere optimerer tjenesteplaner til S-tog

Umiddelbart skulle man måske tro, at Forsvarets Forskningstjeneste kun var noget med krudt, kugler og kampvogne, men som denne artikel viser, kan forskernes viden også bruges andre steder.

Som noget nyt er Forsvarets Forskningstjeneste (FOFT) de seneste år begyndt at påtage sig konsulentopgaver for virksomheder uden for det danske forsvar. Tanken om, at FOFT kunne engagere sig i en sådan konsulentaktivitet, er måske ikke så fjern, da FOFT i sit daglige virke på mange områder optræder som konsulent over for Forsvarets myndigheder og institutioner.

Denne nye aktivitet blev aktuel efter det seneste forsvarsforlig. Der skete her en reduktion i antallet af forskerstillinger ved FOFT, men Forsvarskommandoen har givet FOFT tilladelse til at bevare et antal af disse stillinger, såfremt disse kan finansieres gennem indtægtsdækket virksomhed, dvs løsning af opgaver mod betaling for virksomheder uden for det danske forsvar.

Herved styrkes den faglige kompetence, som Forsvaret erfaringsmæssigt trækker på. Naturligvis må FOFT så anvende ressourcer på at opdyrke og gennemføre indtægtsdækket virksomhed; men denne virksomhed skulle gerne ud over indtægter resultere i et overskud målt i rådigt forskertid, som kan investeres i supplerende forskning og udvikling.

FOFT har allerede i et antal tilfælde udført opgaver inden for kategorien indtægtsdækket virksomhed. En af disse opgaver er emnet for denne artikel.



Tjenesteplaner til S-tog

Opgaven er stillet af DSB S-tog og går ud på at udarbejde planer for lokomotivførernes arbejde. En lokomotivførers opgaver i løbet af en arbejdsdag kaldes en "tjeneste". Af en tjeneste fremgår det bl.a. hvilken togstamme, lokomotivføreren skal køre, og i hvilken tidsperiode. En tjeneste er normalt sammensat af 2 eller 3 perioder, hvor der køres - "frem-

føres"-S-tog, med pause(r) imellem. Desuden en (kort) periode til "check in" og "check out" ved start og afslutning på arbejdsdagen. De fleste lokomotivførere starter og slutter deres tjenester på Københavns Hovedbanegård (KH). Der er overenskomstmæssige regler for, hvor længe en lokomotivfører højst må fremføre tog, inden han skal have pause, og hvor lang, pausen mindst skal være. Der er også en øvre grænse



for, hvor lang pausen må være. Pauser holdes altid på KH, hvor der findes et lokale, hvor lokomotivførerne kan opholde sig. Den længste varighed af en tjeneste er overenskomst-mæssigt fastsat. På den anden side skal det også tilstræbes, at en tjeneste ikke er for kort, da lokomotivføreren altid skal have betaling for et mindste antal timer pr dag. Korte tjenester rundes så at sige op til dette mindste antal timer. I øvrigt skal det tilstræbes, at tjenesterne i gennemsnit har en given varighed.

Udarbejdelse af en tjenesteplan, altså en plan, der dækker alle lokomotivføreres tjenester, er særdeles kompliceret. Hidtil er dette arbejde blevet udført manuelt. Det varetages af DSB S-togs specialister og kræver stor erfaring og megen tid. Mulighederne for hurtigt at lave ændringer i planerne med den eksisterende metode er derfor begrænsede.

De nuværende tjenesteplaner er lavet ud fra den forudsætning, at en lokomotivfører i løbet af en arbejdsdag kan skifte vilkårligt mellem alle linierne. Dette bevirker, at når der skal laves ændringer i planen som f.eks. i tilfælde af sporarbejder, vil ændringerne i princippet berøre hele S-togsnettet (alle tjenester). Dette kunne undgås, hvis planerne var lavet sådan, at en lokomotivfører i løbet af en tjeneste kun kørte på en enkelt linie ("fingerkørsel") eller nogle få linier ("bunkekørsel"). Planerne for de linier, der ikke berøres af sporarbejdet, behøver så ikke at ændres, hvilket vil lette planlægningsarbejdet betydeligt. DSB S-tog har derfor været interesseret i at få udarbejdet planer, der er baseret på princippet for fingerkørsel og bunkekørsel og har indgået kontrakt med FOFT om i et antal udvalgte tilfælde at udarbejde sådanne planer. Opgaven udførtes ved Afdelingen for Operationsanalyse og Matematisk Statistik (ORS).

Når en plan skal udarbejdes, er det naturligt at tilstræbe, at

lokomotivførernes arbejdstid udnyttes bedst muligt. Herved forstås, at lokomotivførerne anvender så stor en procentdel af deres arbejdstid som muligt til togfremføring, altså at togfremføringsprocenten maximeres.

Den optimale plan i en given situation er altså den, der giver den højeste togfremføringsprocent, samtidig med at alle regler vedr. arbejdstid, pauser m.v. er overholdt.

Antallet af mulige planer, selv i tilfældene med fingerkørsel eller bunkekørsel, hvor der ikke er så mange togstammer, bliver imidlertid astronomisk stort. I et af de betragtede eksempler med fingerkørsel vurderes det, at antallet af mulige planer er af størrelsesordenen 10^{50} . Det er klart umuligt at gennemse alle disse planer. Man kan derfor ikke være sikker på at finde frem til den optimale plan, men må være tilfreds med en (nær)optimal plan.

FOFTs opgave kan herefter formuleres som: At bestemme (nær)optimale tjenesteplaner i et antal udvalgte tilfælde med bunkekørsel og fingerkørsel.

Opgaven er særdeles kompliceret og udgør - fra et operationsanalytisk synspunkt - en stor faglig udfordring.

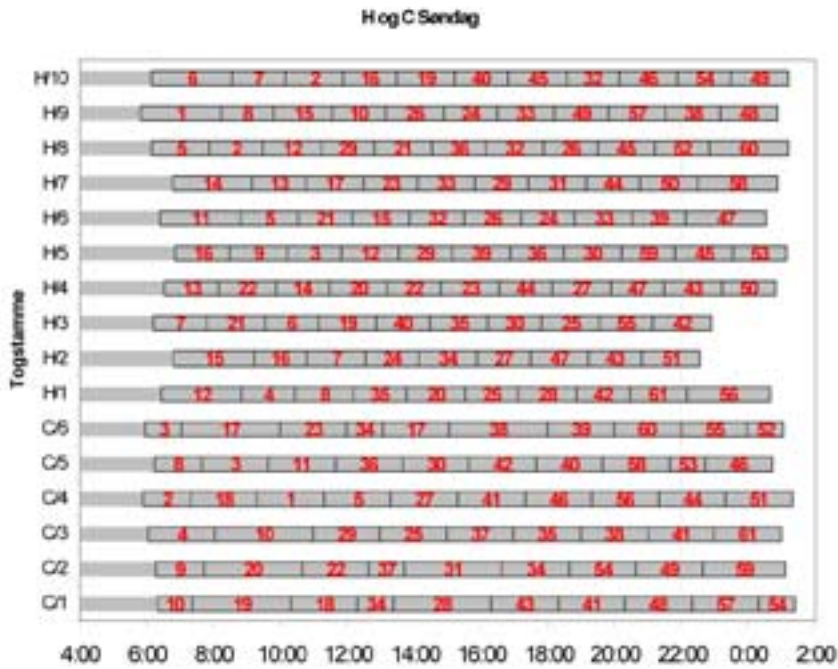
Modeller

FOFT har udviklet to modeller til generering af tjenesteplaner.

- En Branch&Bound metode, som systematisk gennemsøger mængden af mulige planer. På grund af det store antal planer har det ikke været muligt at sikre, at den optimale plan findes. Modellen vil da præsentere den bedste løsning blandt de fundne. Ved at sætte en række parametre er det muligt at lade metoden søge i bestemte dele af mulige løsninger.
- En Heuristisk metode, som udarbejder planer baseret på et regelsæt, der er udviklet ved analyser af eksisterende planer og ved læring fra de planer, der er genereret ved løsning af opgaven. Denne metode gennemsøger ligeledes et stort antal planer. Metoden sikrer heller ikke optimalitet, men genererer meget gode løsninger.

Resultater

FOFT har produceret et antal forskellige løsninger i hver af de udvalgte situationer. Et eksempel på en tjenesteplan for linierne H og C om søndagen er illustreret grafisk i figuren.



Der er i alt 6 togstammer på linie C og 10 togstammer på linie H om søndagen. For hver togstamme er vist en række kasser, der beskriver tilrettelæggelsen af en togfremføring. I kasserne er anført numre på tjenester. Kassernes placering giver start- og sluttidspunkterne for togfremføringerne. Togfremføringerne for hver enkelt lokomotivfører kan findes ud fra figuren. Lokomotivføreren, der varetager tjeneste nummer 2, starter således med at føre togstamme C/4 og efter en pause føres togstamme H/8 og efter yderligere en pause føres H/10, hvorefter arbejdsdagen afsluttes.

Figuren afslører problemets kompleksitet. Løsningen af problemet går ikke blot ud på at placere numre i kasserne, men også ud på at bestemme antallet af kasser og grænserne mellem de enkelte kasser. Sammenlignes fremføringsprocenterne i FOFTs løsninger med fremføringsprocenten i den nuværende tjenesteplan, findes at procenten i FOFTs løsninger er noget højere. Dette kan næppe siges at være overraskende, da FOFT med sine modeller har udnyttet muligheden for at generere og undersøge et meget stort antal planer, en mulighed man på grund af tidsforbruget ikke har ved den manuelle metode. På grund af det store antal

lokomotivførere, betyder hver procentpoints forøgelse af fremføringsprocenten en betydelig økonomisk besparelse.

Det skal bemærkes, at spredningen på tjenesternes længde er noget større i FOFTs løsninger, end i den nuværende plan, selv om den gennemsnitlige længde af tjenesterne i begge tilfælde ligger omkring det ønskede.

Det kan yderligere nævnes, at FOFT som en ekstra opgave har udarbejdet tjenesteplaner for strækningen Hillerød – Helsingør (»Lille Nord«). Også her har FOFT opnået planer med en højere fremføringsprocent end i den nuværende plan.

Afslutning

FOFT har i henhold til kontrakten med DSB S-tog haft flere opgaver end de her omtalte og forventer, at samarbejdet med DSB S-tog vil fortsætte.

Denne artikel er oprindeligt bragt i FOFT-nyt august 2002. DSB S-tog a/s har velvilligt stillet billedmateriale til rådighed.



Af Torben Christensen (email: tc@ddre.dk) og Jens Meng Hansen (email: jmh@ddre.dk) Forsvarets forskningstjeneste, Svanemølle kaserne.

Artiklens forfattere er seniorforskere hos FOFT ved afdelingen for operationsanalyse og statistik.



Forsvarets Forskningstjeneste, der er virksomhedsmedlem af DORS, er et sektorforskningsinstitut, som bistår forsvaret med løsning af problemer af forskningsmæssig karakter. I nationalt og internationalt forskningssamarbejde varetager FOFT forsvarets interesser, ligesom man koordinerer væsentlige forsknings- og udviklingsaktiviteter ved forsvarets myndigheder, indsamler, registrerer, bearbejder og videregiver forskningsdata og -resultater, som skønnes af betydning for forsvaret. FOFT er opdelt i tre afdelinger:

- Sensorafdelingen
- Operationsanalyse- og statistikafdelingen
- Datamatikafdelingen

Ny bog om ruteplanlægning

Den måtte komme på et eller andet tidspunkt, og nu er den her så. Sammen med Daniele Vigo har Paolo Toth redigeret en bog om ruteplanlægningsproblemet. Ruteplanlægningsproblemet og varianter heraf er en absolut klassiskere inden for operationsanalyse. Bogen giver et fornemt overblik, men den er ikke uden problemer.

Stort set samtlige af operationsanalysens "store kanoner" har medvirket i udforskningen af ruteplanlægningsproblemer og varianter heraf, og mængden af forskningsresultater er enorm. Alligevel er det først nu, at der er udgivet en samlende bog om emnet.

Bogen er redigeret af to af Italiens førende forskere inden for operationsanalyse: Paolo Toth og Danielle Vigo. Tilsammen har de mere end 120 publikationer bag sig heraf en betydelig del inden for ruteplanlægning. I bogen har de samlet et team af kompetente og betydningfulde forskere, som skriver om forskellige problemstillinger inden for ruteplanlægning.

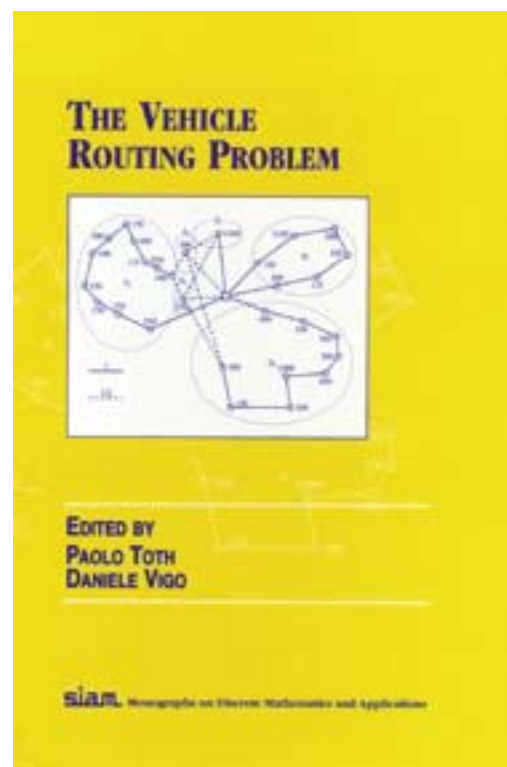
Udgangspunktet for bogen er the Vehicle Routing Problem (VRP) ofte også kaldet the Capacitated Vehicle Routing Problem (CVRP). Næst efter det nærmest legendariske rejsende sælgers problem (the Travelling Salesman Problem - TSP) er CVRP det simpleste ruteplanlægningsproblem. TSP ville nok have været det mest logiske udgangspunkt for en bog om ruteplanlægning, men da der allerede findes bøger om dette emne, har man klogeligt valgt CVRP som fundament. Den operationsanalytiske behandling af ruteplanlægningsproblemer har mere end 40 års historie bag sig. Tilbage i 1959 beskrev Dantzig og Ramser et

problem med udbringning af benzin til tankstationer. De opstillede en matematisk model og en algoritmisk løsningsmetode. Udforskningen af CVRP tog dog først for alvor fart i forbindelse med at kraftfulde IT-systemer blev billige og alment tilgængelige.

CVRP kan beskrives ganske enkelt. Givet er et depot og en række kunder. Fra depotet forsynes kunderne med varer via en flåde af lastbiler. I den grundliggende model er alle lastbilens, og opgaven består nu i at minimere den samlede omkostning ved udbringning af kundernes varer. Alle kundernes ønsker skal opfyldes, og leveringen hos en kunde skal ske med præcis én lastbil ligesom lastbilerne har en kapacitetsbegrænsning. Så simpel denne model end måtte lyde har den været et vigtigt akademisk instrument til at undersøge forskellige løsningsmetoder inden for ruteplanlægning.

Bogen er delt op i 3 dele: Første del omhandler CVRP med kapitler om eksakt og heuristisk løsning af problemet. I anden del gennemgås de tre vigtigste varianter af VRP: VRP med tidsvinduer, VRP med backhaul samt Pickup-and-Delivery. Tredje del omhandler anvendelser og case studies.

Beskrivelsen af eksakte metoder



"The Vehicle Routing Problem" af Paolo Toth og Daniele Vigo. ISBN 0-89871-498-2

Udgivet af Society for Industrial and Applied Mathematics (SIAM). Vejl. pris: 95 USD

inden for CVRP er opdelt efter de tre hovedmetoder, der har været anvendt: Branch-and-bound, branch-and-cut og set-partitioning. Derudover er der 2 kapitler om heuristikker til løsning af CVRP, bag hvilke professor Gilbert Laporte er en af hovedkræfterne. Denne del af bogen er den absolut stærkeste. Stoffet er i denne del frem-

ragende disponeret og professionelt skrevet, og specielt de to kapitler om heuristikker giver læseren god mulighed for at danne sig et overblik over et stort område bestående af mange forskellige heuristikker.

Bogens anden del gennemgår de tre mest udforskede specialiseringer af CVRP: VRP med tidsvinduer, VRP med backhaul og Pickup-and-Delivery. For den absolutte kender vil den særdeles korte beskrivelse af heuristikker til VRP med tidsvinduer være iøjnefaldende. Arbejdet inden for dette område har været særdeles aktivt siden slutningen af 80'erne og har resulteret i en stor mængde litteratur. Forskningsaktiviteten inden for heuristiske metoder har været særdeles høj, men dette fremgår desværre ikke af bogens indhold. Kapitellet er betydelig svagere end kapitlerne i den første del af bogen.

Den tredje del indeholder fem meget forskellige kapitler om anvendelse af ruteplanlægning. Specielt kapitlerne "Routing Vehicles in the Real World" og "Evolution of Microcomputer-based Vehicle Routing Software" er særdeles interessante for både industri og universitetsverdenen. Foruden at være godt skrevet rummer kapitlerne gode grunde til at interessere sig for de mere avancerede sider af ruteplanlægning og ikke kun de gængse simple indsættelsesmetoder, som industrien i mange år har haft vanskeligt ved at løsrive sig fra. Kapitlerne burde desuden give forskere incitament til at søge kontakt med industrien.

Bogens styrke ligger i, at den er skrevet af forskere med stor erfaring og ekspertise inden for problemstillingerne. Forfatterne er alle at regne blandt den internationale elite inden for området. Dette er desværre også en af bogens store svagheder. Selvom om Toth og Vigo givetvis har gjort et stort arbejde

med at koordinere indholdet af de enkelte kapitler, er det et irriterende moment med forskellige notationer igennem bogen, værst er det dog, at bogen ikke evner at trække de store linier op på tværs af kapitlerne. Kapitlerne er overordnet set selvstændige enheder, og dette kommer bogen desværre aldrig ud over. Bogen kunne med fordel bruges i forbindelse med et kursus i ruteplanlægning eller transportoptimering, men med en pris omkring 700 kr. er dette næppe muligt, idet den vanskeligt kan stå alene som lærebog.

"The Vehicle Routing problem" er et godt og solidt værk om et af operationsanalysens klassiske problemstillinger. Skrevet af en stor skare af kompetente forskere inden for området er denne bog et sikkert opslagsværk for folk, der arbejder indenfor området og jeg tror at både universitetsfolk og operationsanalytikere ude i industrien kan få inspiration til nye projekter af denne bog.



Af Jesper Larsen
Information og Matematisk
Modellering, Danmarks Tekniske
Universitet, e-mail: jla@imm.dtu.dk.

Faglig beskæftiger Jesper sig primært med ruteplan- og mandskabsplanlægning samt forskellige operationsanalytiske problemstillinger indenfor luftfartsindustrien. Derudover er han den ene af to redaktører op ORbit.

Abstracts i operationsanalyse

DORS' bestyrelse har udpeget Jesper Larsen (en af ORbits to redaktører) til at være »contributing editor« (CE) for Danmark for tidsskriftet »International Abstracts in Operations Research« (IAOR). IAOR samler abstracts om operationsanalyse fra forskellige kilder (via CE'er i alle lande der er medlem af IFORS) og udgiver dem redigeret som en slags opslagsværk.

Opgaven som CE er at holde øje med artikler inden for forskning og udvikling i operationsanalyse i Danmark. Så hvis du/I publicerer en interessant artikel inden for operationsanalyse, bedes I kontakte Jesper så han kan rapportere videre til IAOR.

Det drejer sig i den forbindelse specielt om materiale, der bliver udgivet i tidsskrifter, som normalt ikke indeholder operationsanalytiske artikler. De sædvanlige »mainstream« tidsskrifter inden for operationsanalyse bliver løbende evalueret af IAOR.

På DORS' hjemmeside kan du se, hvilke informationer Jesper gerne vil have for at kunne reportere til IAOR. Du kan kontakte ham via email på jla@imm.dtu.dk, eller pr. brev IMM, DTU - bygning 305, 2800 Kgs. Lyngby. [jla]

DORS' 40 års jubilæum fejret i november

Den 21. august 1962 blev Dansk Selskab for Operationsanalyse stiftet. Selskabet kunne således sidste år fejre 40 års jubilæum. Bestyrelsen havde derfor forberedt et jubilæumsarrangement, der ikke kun kiggede tilbage men også frem.



Direktør Ole Kessel fra Transvision er en særdeles erfaren mand inden for operationsanalyse. I hans foredrag gennemgik han mange af de projekter han i tidens løb har været med til at forme.

Ved en festlig sammankomst den 7. november sidste år fejrede selskabet således sit 40 års jubilæum i den smukke Alexandersal på Københavns Universitet. Meningen med arrangementet var ikke kun at se tilbage men i lige så høj grad at se fremad.

LM Systems ApS

I den ånd var Jesper Leedgaard og Kim H. Mortensen inviteret til at give aftenens første foredrag. Kim og Jesper har gjort deres eksamensprojekt fra Danmark Tekniske Universitet til deres levevej. I deres eksamensprojekt arbejdede de med mandskabsplanlægning for detailhandlen (er du interesseret



DORS' formand Søren Nielsen byder velkommen til jubilæumsarrangementet.

kan du læse mere herom i de to artikler fra DORSnyt nr. 121 og 122). Foruden en gennemgang af deres system kom Kim og Jesper også ind på mange af de aspekter, der skal tages hensyn til når man søger at sælge et system der er unikt og innovativt indenfor dets område.

Ole Kessel

Aftenens andet foredrag blev givet af Ole Kessel, direktør i firmaet Transvision, der har specialiseret sig i rute- og transportplanlægning. Ole har om nogen fulgt og præget dette område i Norden siden han stoppede sit ph.d.-studium for at forfølge sine interesser kommercielt.

Erik Johnsen

Aftenens hovedforedrag stod professor Erik Johnsen for. Forinden var han, som annonceredes i sidste nummer af ORbit blevet udnævnet til det andet æresmedlem af DORS. Sammen med Arne Jensen, som er det første æresmedlem, var Erik Johnsen afgørende for oprettelsen af DORS.

Erik Johnsen, der midt i 50'erne, lærte om operationsanalyse i USA har tilbragt stort set hele sin aktive forskningskarriere på Handelshøjskolen i København, hvor han stadig er aktiv ved CAMS (Center for Advanced Management Studies). [jla]



Professor Erik Johnsen, som stod for aftenens hovedforedrag, blev ved samme lejlighed udnævnt til æresmedlem for hans store engagement i udbredelsen af operationsanalyse, samt hans pionerånd inden for talrige områder.

Målstyrning, operationsanalys och multimål

Målstyrning har blivit aktuellt i dagens utbildnings-Sverige. Kan det fungera för utbildning och forskning som är komplexa verksamheter med många olika mål? »Om det skall vara någon rimlig mening med att åstadkomma bättre företagsstyrning, så måste detta även innebära att man styr så att resultaten som uppnås blir bättre ur de högsta målens synpunkt och ej blott enligt traditionella produktivitetsschabloner, som på sin höjd kan motsvara någon bråkdel av den totala, högsta målstrukturen.«

Så skriver Sveriges förste professor i Informationsbehandling, Börje Langefors, i inledningen till sin bok »System för företagsstyrning«. Han har arbetat med frågan om mål i flera olika sammanhang, dels under sin tid vid SAAB och dels vid universitetet där målbegreppet kom att spela en central roll i arbetet med »administrativa informationssystem«. I denna text beskriver han bakgrunden till utvecklingen av målstyrning och visar hur det i de flesta organisationer finns flera olika mål. Studiet av »multimål-situationer« är därför en viktig del av operationsanalys.

Målens betydelse

Mål är viktiga i många olika situationer. När man skall styra måste man ha gjort klart vilket mål man skall styra mot. När man skall fatta beslut, måste man veta vart man syftar; mot vilket mål man strävar. För att motivera människor att anstränga sig, måste man göra klart för dem vad målet är och för att värdera kvalitet måste man ställa egenskaper i förhållande till mål. Det är förvånande att människor ofta är ovetande om målens betydelse och oklara om vad, som menas med mål, fastän mål är av utslagsgivande betydelse i många olika sammanhang. Det förekommer inte sällan att man förnekar att ledare eller

organisationer alls har några mål. Detta gäller inte bara vid samtal, utan även i litteraturen.

Beslutsteori, optimering och målfunktioner

Under andra världskriget anlätades vetenskapsmän och matematiker i planering och operationer, för att söka finna effektiva eller optimala lösningar på olika militära, operativa problem. Det framkom på detta sätt ett antal problemtyper, av ganska allmän karaktär, för vilka metoder för att finna optimala lösningar kunde utvecklas. Detta verksamhetsområde kom att kallas »operationsanalys«.

Svenska matematikersamfundet började under 1950-talet söka efter företag, som var villiga att lämna uppgifter för försök att industriellt utnyttja operationsanalysen. Vid nästan alla operationsanalytiska tillämpningar handlar det om att optimalt tillgodose en målfunktion. Tanken är att om bara uppdragsgivaren kan ange sitt mål så kan sedan metoden ange den optimala lösningen.

Uppdragsgivaren kan på så sätt, genom målformuleringen, styra beslutet, utan att behöva förstå metoden i dess detaljer. Operationsanalysen blev därigenom

också ett mönster för delegering. Delegering var något, som jag ofta kom att sakna under min tid på SAAB.

Målbegreppet vid tekniskt konstruktionsarbete

Vid ett tillfälle, under min tid vid SAAB, kom målbegreppet att ytterst handgripligt presentera sig för oss, i samband med teknisk konstruktion. Vi hade gjort stora framsteg med automatisering av hållfasthetsberäkningar för flygplan och började att fundera över vad, som skulle vara det mest angelägna nästa framsteget. Man borde nu kunna ta ett steg mot att automatisera konstruktionsarbetet.

Hållfasthetsberäkningarna resulterar i en förteckning över materialpåkänningarna i olika delar av flygplansstrukturen. Mot bakgrund av denna information fattar konstruktören beslut om att ändra materialets fördelning. En balk kanske skall göras starkare och några andra klenare, för att flygplanet skall kunna tåla större belastning eller klara samma belastning med mindre vikt.

Det framstod som möjligt att göra datorprogram, som systematiskt gjorde en serie förutbestämda typer av omfördelning av material. För varje ny fördelning

skulle då materialpåkänningarna och vikten beräknas. Efter varje sådan modifiering skulle datorn (programmet) beräkna det resulterande värdet på kvaliteten. Det gällde att söka det bästa värdet. Vid denna punkt blev det klart att man måste kunna definiera ett kvalitetsmått. Det verkade naturligt att tänka sig ett mått av typen minsta vikt vid en förutbestämd högsta tillåtna materialpåkänning. Detta blir då ett mål som konstruktören skall söka uppnå.

Nu började svårigheterna. Fördelningen av materialpåkänningarna varierar vid olika flygsituationer. Vid en flygsituation blir en viss del av planet mest ansträngd, vid ett annat tillstånd en annan del. Detta medför då att den bästa materialfördelningen blir olika vid olika »flygfall«. Här mötte vi nu inte bara problemet att definiera ett må att uppfylla eller närma sig till, det blev frågan om flera sådana mål att (maximalt) uppfylla i en och samma konstruktion, ett mål för varje flygfall. Det var uppenbart att detta inte var möjligt. Istället måste man fråga sig om man kunde bestämma en bästa möjliga kompromiss mellan de olika flygfallen. Vi hade alltså kommit fram till en »multimål-situation«, även i ett relativt rendlat tekniskt problem. Detta var något, som operationsanalysen helt tycktes ha blundat för, och som blir mycket mer påtagligt i samband med styrning av organisationer.

Administrativa informationssystem och mål.

Företagsledning och administration har till uppgift att få företaget eller organisationen att fungera effektivt. För att förstå vad detta innebär måste man klargöra vad man menar med effektivitet. Företagsfilosofen Peter Drucker påpekade att det engelska ordet efficiency behöver kompletteras med ett annat begrepp: effectiveness, som

skulle utgöra en värdering av i vilken utsträckning man gör de rätta sakerna: »An effective executive is a man who gets the right things done«. Efficiency har istället att göra med output i förhållande till input.

Med effectiveness i betydelsen att göra de rätta sakerna, uppstår frågan om vad som är de rätta sakerna. Drucker hade ett klart svar på den frågan. Man är effektiv om man uppfyller målen. Men viktiga analytiska såväl, som praktiska frågor uppstår, när man vill gå närmare in på dessa begreppsbyggnader: att uppfylla målen är inte att göra de rätta sakerna om man inte gjort säkert att målen är de rätta och vidare; om man inte helt uppfyller alla målen, vad är då effektiviteten? Om av två alternativa sätt att lösa en uppgift, den ena fyller ett av målen medan det andra uppfyller ett annat, vilket har då högst grad av måluppfyllelse?

När jag tillträdde professuren i »Administrativa Informationssystem« bedömde jag att en central uppgift för forskningen om informationssystem för företag borde vara att nå kunskap om hur informationssystem kan medföra bättre fungerande och bättre ledda organisationer. Då uppstod behovet att ange hur man skulle kunna mäta hur effektiviteten av organisationen förändras genom användandet av ett informationssystem. Detta var ju nödvändigt om man ville utveckla metoder för att göra goda informationssystem. Därför behövdes det skärpta effektivitetsbegreppet.

Informationssystem för Styrning.

Arbetet med optimeringsteorier hade gett antydning om att med mera datorkraft och kraftfullare matematiska metoder - båda rimliga framtidsväntningar - skulle det kanske tom bli möjligt att utveckla informationssystem,

vilka kunde styra företagen optimalt. Det blev då en naturlig första etapp i forskningen, att pröva huruvida detta vore ett realistiskt perspektiv. Om så ej vore fallet vore det också värdefullt att bevisa det, för att undvika omöjliga forskningsinsatser.

Det visade sig förvånande snabbt att tanken på totaloptimering var fullständigt orealistisk, oavsett hur stora framsteg datortekniken skulle komma att göra. Det fanns åtminstone tre olika skäl till detta, vart och ett tillräckligt: 1. företag har flera mål, som dessutom är odefinierade, i början. 2. det finns inte och kan inte finnas tillräcklig kunskap om företag för att med säkerhet förutse vad som blir effekten av en styråtgärd, och 3. det går inte, även med aldrig så kraftfulla datorer, att få tillgång till all den information, som skulle behövas - och få den i tid. Det är alltså omöjligt att totaloptimera ett företag.

Organisationer har flera mål

Det är en vanlig föreställning att företag har ett enda mål: att åstadkomma största möjliga vinst. Inte bara allmänheten, utan också företagsledare och nationalekonomer brukar ha detta som en utgångspunkt. Vid den tid, då jag började studera problemet företags mål började det förekomma argument i den tongivande företagsekonomiska litteraturen, som framhöll att företag har flera mål. Detta är besvärande på flera sätt: det blir inte längre möjligt att på vetenskapliga, logiska grunder bestämma vilka mål, som är de högsta och man kan därför inte heller ange en teoretiskt bästa målstruktur. Det blir en fråga för bedömning och förhandlingar att bestämma de högsta målen och deras inbördes vikter.

Det är egentligen ganska lätt att inse att företags beslut kan inrymma flera olika

mål och att dessa inte kan bestämmas på objektiv, vetenskaplig väg, utan måste grundas på bedömningar och beslut. Några av de mest typiska företagsekonomiska besluten har att göra med investeringar:

att avstå något nu, för att få ut något av framtiden. När man talar om största möjliga vinst måste man därför alltid ta ställning till hur mycket

man skall minska vinstuttaget idag, för att också kunna få vinster nästa år och nästa igen. Och ändå brukar det påstås att företagen har inget annat mål än att maximera vinsten. En diskussion, som jag för många år sedan hade med en holländsk professor, tidigare teknisk direktör vid Philips, kan belysa föreställningarna om företagets vinstmål. Jag berättade för honom om mitt intresse för multimålsproblemet och att detta innebar att man inte kunde veta vilket, som var företagets högsta mål. Hans reaktion var att detta inte var något problem; företag hade ju enbart målet att ge största möjliga vinst. På min retoriska fråga: »varför skall företag ge största möjliga vinst gjorde han en min, som tydligt visade att det var en dum fråga. Motvilligt gav han dock ett svar: « t.ex. för att överleva.« - »Men då är ju överlevnad ett mål, som är överordnat vinstmålet, invände jag.« » Och varför måste företaget överleva«? - Det blev mycket tydligt att detta var en otroligt dum fråga. Det kom dock ett svar: »T.ex. för att kunna ge arbete åt folk«. - »Så nu har vi alltså kommit till att arbete är överordnat både överlevnadsmål och vinstmål«. - Nu stod han inte ut längre utan reste sig och gick sin väg, med en vredgad min. Vi kom alltså inte fram till vilket, som skulle kunna vara företagets högsta mål.

I boken »System för Företagsstyrning« undersökte jag olika tänkbara alter-

nativ för entydiga högsta mål, vilka hade angivits i den litteratur, som jag studerade. Ett av dessa var minsta möjliga kostnad. Det var dock rätt lätt att visa att detta inte var en framkom-

»Effektivitet som måluppfyllelse innebär att man bara gör de rätta sakerna om målen är de rätta.«

lig väg. Detta kan illustreras av en dispyt, som jag blev indragen i vid en konferens i Genève, om »Computers in Management«. Jag hade föreläst om problemen med multimål, varvid jag börjat med att påpeka att mitt syfte inte var att visa att företag måste ha flera mål. Detta var väl etablerat i litteraturen och jag sade mig utgå ifrån att detta var känt. Trots det visade jag med det enkla investeringsexemplet, att man lätt kunde inse detta om man inte tidigare kände till det.

Efter mig föreläste en ingenjör från England om produktionsstyrning vid den engelske partnern i Concorde - projektet. »Vi har bara ett enda mål, att styra efter, professor Langefors, det är att minimera kostnaden«. Min replik var då att isåfall fanns det en enkel lösning: att lägga ned projektet. Detta blev en mera dramatisk invändning än jag tänkt mig. Concorde projektet var ju mycket omdebatterat i den internationella pressen. Mannen fick tårar i ögonen och gav en vemodig replik. Tekniskt sett vore förstås invändningen på min replik att kostnadsminimeringen gällde under iakttagandet av vissa »bi-villkor«. Men poängen var att då skulle dessa sidovillkor vara bland de högsta målen, och så är vi tillbaka till multimålsproblematiken.

På senare tid har företagsledare börjat att ersätta talet om största möjliga vinst med »uthållig lönsamhet«. Detta framhäver problemet med framtidsperspektivet och är såtillvida en förbättring, men det gör ju inte något konkret åt problemet med att väga nutid mot framtid och

detta framstår fortfarande som ett problem om bedömningar och värderingar. Det går inte att undvika multimålsproblemet med nya ord.

Multimål var ett besvärande problem. Det innebar ju både att man inte längre kunde veta vilka mål ett givet företag hade (eller borde ha) och att man inte kunde mäta effektiviteten och därför inte heller kunde bestämma optimala beslut. Som jag redan berättat, förekom det tom i ganska väldefinierade tekniska konstruktionsproblem, att konstruktören hade att hantera flera olika mål.

Ultimära mål, önskemål och värderingar.

Effektivitet som måluppfyllelse innebär att man bara gör de rätta sakerna om målen är de rätta. Mål kan förekomma på olika nivåer. Om tex en bestämd marknadsandel på en viss produkt, anges som ett operativt mål, så är det sannolikt att detta sker för att man bedömer att man därigenom skall uppnå en bättre vinst. Om så är fallet så är ju vinstmålet överordnat marknadsandelsmålet. Det framstår nu som tänkbart att flera olika operativa mål skulle ha ett enda ultimärt mål. Då skulle ju problemet med multimålen ha fått en effektiv lösning; man skulle ju då på samma gång ha löst problemet med att bestämma effectiveness. Den skulle ju då vara graden av uppfyllelse av det enda och ultimära målet.

Om man fattar beslut om målen, utan att ha analyserat situationen i alla de olika aspekterna, innebär det tydligen att man söker lösa ett ytterst komplicerat problem på en gång, med enbart hjälp av bedömning. Man har naturligtvis inte rätt att vänta sig att resultatet blir bra. Tyvärr är det ofta ändå så det går till, även i mycket viktiga beslut, tex om rikspolitiken. Ofta refererar politikerna till detta som en fråga om »vär-

deringar», medan det rör sig om en sammanblandning av värderingar och gissningar om vilka åtgärder, som bäst kan leda till uppfyllande av värderingarna. Det är inte underligt att ekonomisk forskning i USA funnit att politiska beslut ofta lett till resultat, som blev de motsatta mot dem man tänkt sig.

För en metodforskare är det naturligt att vänta sig att man skulle få mycket bättre målsättning, om man tar ett analyserande grepp. Detta visar sig också ge anvisning om hur olika aktörer kan bäst utföra olika delar av målsättningsprocessen. Ett bidrag till ett visst önskemål är vanligen mera värt, för intressenterna, än ett lika stort bidrag till ett annat önskemål. De olika önskemålen måste alltså tillmätas olika vikt. Om man kunde ange vikten eller värdet av varje önskemål, så skulle den viktade summan av bidragen från de olika målen, framstå, såsom ett enhetligt totalmål. Det är omöjligheten att bestämma vikterna på objektiv väg, som gör att multimålsituationen är ofrånkomlig.

En del av målsättningsprocessen är av en karaktär för vilken det ofta kan finnas specialister vilka bättre än brukarna kan definiera vad, som kan göras och vilka effekter, som kan väntas av olika åtgärder. Ingen annan än brukarna (i vidaste mening) har däremot en reell möjlighet att fastställa vilka önskemål, som skall tillgodoses. Det är också bara dessa, som kan bedöma vilken vikt, som skall tillmätas de olika önskemålen.



Börje Langefors är professor emeritus i "Informationsbehandling, särskilt administrativ databehandling". Han ledde under 1950-talet som ingenjör vid SAAB utvecklingen av datorn SARA. 1963 blev han docent i byggnadsstatik vid Chalmers på arbeten om finita

elementmetoder. Han är ekonomie hedersdoktor vid Lunds universitet 1975, filosofie hedersdoktor vid Göteborgs universitet 1987 och teknologie hedersdoktor vid KTH 1989. 1999 tilldelades han "LEO AWARD for Lifetime Exceptional Achievement in Information Systems". Hans publikationer omfattar många olika områden, t.ex.

- "Algebraic Topology for Elastic Networks"; Saab, TN49, Linköping 1961
- "Dynamique des Fluides" i "Manuel de Base de l'Ingenieur", DUNOD; Paris 1959
- Theoretical Analysis of Information systems (THAIS), Studentlitteratur, 1966
- System för företagsstyrning, Studentlitteratur, 1968
- Essays on Infology (ed. Bo Dahlbom) Studentlitteratur, 1995

Denne artikel er oprindeligt bragt i Fysikaktuelt nr. 3 2002. Fysikaktuelt er Svenska Fysikersamfundets medlemsblad.



Produktionsplanlægning hos Tulip med SAS/OR

Tulips skinkedivision i Brabrand har opnået store besparelser i deres processtyring ved at gå ind og analysere på de enkelte dele af processerne med softwarepakken SAS/OR fra SAS Institute. Med SAS/OR til operationsanalyse har Tulip været i stand til at lokalisere områder, som kunne optimeres. Her beskrives, hvordan SAS/OR er blevet en del af produktionsplanlægningen hos Tulip.

På Tulips skinkedivision i Brabrand produceres en række forskellige færdigvarer (skinkeprodukter mm., her kaldet saltvarer). De producerer omkring 600 tons pr. uge, dvs. 30.000 tons pr. år. Den samlede årlige omsætning er omkring 1 milliard kroner, og langt størstedelen af udgifterne stammer fra køb af råvarer. En effektiv udnyttelse af råvarerne giver derfor store besparelser.

Eksempler på brug af SAS/OR hos Tulip

Den langsigtede planlægning går ud på at finde de optimale recepter, så de ønskede saltvarer produceres billigst muligt (indkøbsstyring). Planlægningshorisonten er her typisk en måned.

Den kortsigtede planlægning (produktionsstyring) har samme mål, men her kan recepten for den enkelte vare naturligvis variere fra dag til dag. Denne



planlægning kompliceres ved, at overskydende kød kan gemmes, idet det dog senest skal bruges dagen efter, at det er indkøbt. Endelig skal der tages hensyn til randbetingelser, dvs. startbeholdninger samt ønskede slutbeholdninger af skinkeråvarer.

På langt sigt (1-2 mdr.) fastlægger man, hvor meget der i alt skal produceres af hver saltvare, og på kort sigt (1-2 uger) fastlægges desuden, hvordan den samlede produktion skal fordeles ud på planlægningsperiodens enkelte dage.

Til produktion af hver saltvare medgår et antal råvarer med forskellige priser. Råvaresammensætningen for saltvarerne fremgår af såkaldte 'flydende recepter', hvori der angives øvre og nedre grænse (kan evt. være sammenfaldende) for hver enkelt råvares procentuelle andel i saltvaren.

Visse af råvarerne stammer fra velfærdsgrise og kaldes derfor velfærdsråvarer, svarende til at visse af saltvarerne er velfærdsprodukter. I recepterne for disse indgår naturligvis kun velfærdsråvarer, men hvis der er velfærdsråvarer til overs må disse godt anvendes i almindelige produkter.

Råvarerne inddeles i A- og B-råvarer og til hver enkelt saltvare knyttes en minimumsprocent, der angiver hvor stort procentuelt indhold af A-råvarer, der mindst skal være i saltvaren. Samtidig indgås der aftaler om køb af et fast antal hele skinker (velfærds- og almindelige skinker) for perioden (evt. pr. dag). En skinke vejer godt 6 kg. og har en bestemt sammensætning af 10 af de mest benyttede råvarer. For nogle af disse råvarer gælder det, at der ikke kan indkøbes mere end det, der sidder på skinkerne, hvorimod de resterende – samt alle de råvarer, der ikke sidder på skinken – kan suppleres ubegrænset op.

Tulips farsdivision

Tulips farsdivision i Vejle har også benyttet operationsanalyse til optimering af deres processer. Farsdivisionen har en årlig produktion i størrelsesordenen 35.000 tons farsprodukter (primært dåseskinke, leverpostej og pølser).

Også i dette tilfælde er målet at effektivisere de processer, som udgør en stor del af omkostningerne. Processerne her er nogenlunde de samme som hos Tulip skinkedivision om end med en del variationer.

F.eks. er fedtindholdet i de færdige varer meget vigtigt her, og derfor har hver råvare tilknyttet en fedtprocent og hver færdigvare et interval, inden for hvilket den resulterende fedtprocent skal ligge. Endvidere kan de samme varer indkøbes fra forskellige leverandører (med forskellige priser og forskellige leverancebegrænsninger). Til gengæld bortfalder her problemstillingen med de hele skinker samt muligheden for at gemme råvarer til næste dag.

Til farsdivisionen er implementeret tre løsningsmodeller:

Budgetoptimering (tidshorisont 1 måned): Kørslen danner grundlag for indledende forhandlinger med leverandørerne, hvorefter processen gentages flere gange.

Indkøbsoptimering (tidshorisont 1 uge): Her fastlægges præcis hvad der købes hvor og hvornår.

Blandingsoptimering (hver nat): Her er råvarerne til stede og er fordelt i bakker á ca. 30 kg. Problemet er at bestemme hvilke bakker, der skal sendes til hvilke produktionslinier, så puslespillet går op. Her er egentlig tale om et heltalsprogrammeringsproblem, men af forskellige grunde har Tulip valgt at finde den optimale ikke-heltals løsning, afrunde denne til en heltalsløsning, afrapportere konsekvenserne og herefter hjælpe brugeren igennem en manuel iterationsproces, der fører til den foretrukne løsning. Denne eksporteres herefter til det operationelle system, der styrer selve produktionen.

Succesfuld implementering

En af de vigtigste betingelser for en succesrig implementering af et sådant system er, at brugerne deltager aktivt i udviklingen. Specielt er det vigtigt i de indledende faser, hvor problemet skal formuleres og afgrænses. Gennem diskussioner af problemstillinger får man øjnene op for rutiner, der hidtil har været gennemført på uhensigtsmæssig vis samt ukorrekte data.



Af Per Allan Jensen (e-mail: per.allan.jensen@sdk.sas.com), SAS Institute Danmark.

Per Allan Jensen er chefkonsulent i SAS Professional Services.



Billederne er venligst stillet til rådighed af Tulip og Danish Crown



Tulips skinkedivision producerer 30.000 tons skinker om året på et marked, hvor pris og kvalitet er ufravigelige konkurrenceparametre. Da Tulip ikke ønskede at gå på kompromis med kvaliteten var optimering af de interne processer midlet til at forblive konkurrencedygtige på prisen. Tulip har derfor benyttet operationsanalyse til at optimere deres processer.

Forretningsløsningen

Operationsanalyse i form af matematiske modeller af 'det virkelige liv' bliver ofte meget store og komplekse. En del softwareleverandører forhandler programmer, der kan løse sådanne problemer, bl.a. SAS Institute med pakken SAS/OR. SAS's styrke i forhold til andre leverandører er, at de - ud over at have de nødvendige OR-løsningsmodeller - råder over en lang række komponenter til brug for implementering af en skræddersyet og køreklar implementering hos kunden: Access til alle gængse databaser, rensning, kontrollering og klargøring af data, dataentry, applikationsoverbygning (lokal, client-server eller inter-/intranet) og afrapportering i form af grafer eller rapporter (og/eller eksport af data til de operationelle systemer). Derudover råder SAS over en stor stab af konsulenter med bred forretningsforståelse samt stor erfaring i at løse komplekse problemstillinger og styre projekter.

Constraint Programming versus Mathematical Programming

Constraint Logic Programming (CLP) is a relatively new technique from the 80's with origins in Computer Science and Artificial Intelligence. Lately much research have been focused on ways of using CLP within the paradigm of Operations Research (OR) and vice versa. The purpose of this paper is to discuss the differences and similarities between the two mentioned paradigms and how they can complement each other.

In OR and Mathematical Programming we are modeling and solving hard combinatorial optimization problems with use of Linear Mixed Integer Programming models (MIP). *Programming* in MIP has nothing to do with *computer programming*. Rather the result is a program or schedule for doing the optimized activities. In CLP on the other hand a constraint program is both a statement of a problem in variables and constraints and a specification of how to solve it. A CLP system contains a programming language component and a *constraint store* component that can store and reason about variables and constraints.

In the OR community we are interested in formulating and solving models of the following type:

$$\min z = \sum_{j=1}^m c_j x_j \tag{1}$$

$$\sum_{j=1}^m a_{ij} x_j = b_j \quad i = 1, \dots, n \tag{2}$$

$$x_j \geq 0 \quad j = 1, \dots, m \tag{3}$$

$$x_j \text{ integer} \quad j = 1, \dots, m \tag{4}$$

A wide range of applications can be modeled and solved within this paradigm, but some are not well suited. The objective function and the constraints have to be formulated as linear functions, which is not always simple to do.

CLP on the other hand can handle non-linear functions in any type of constraint, for example $x^2 \neq y$. The increase in expressiveness when modeling is the major motivation to take a closer look at CLP from an Operations Researcher's point of view.

The CLP system *ECLiPSe* is a very flexible environment

for implementing and solving both MIP, CLP and combined models, which we will use in the paper to illustrate the different possibilities. *ECLiPSe* is based on Prolog, but we will only describe specific Prolog constructs, which are needed for our purposes. Clocksin and Mellish [2] is the classic, but perhaps a bit out-dated book on Prolog.

Before considering the modeling possibilities in CLP we shall in section 2 recall the components of Branch & Bound for solving MIP problems and compare that to the CLP approach. In section 3 we show an example of modelling and solving a problem with both a MIP and a CLP model in *ECLiPSe*.

The potential of both paradigms is even more evident when they are combined into *hybrid solvers*, which is however not covered in this article. We have the intention to write a part two of this paper, covering hybrid solvers and CLP in more depth.

Comparing Branch & Bound and CLP

Assume we are to minimize the objective function, $f(x)$ given by (1) over the region of feasible solutions, $x \in S$ given by equations (2)-(4). Let z^{opt} be the optimum value.

Branching and Tree Search

In order to guarantee finding the best solution in Branch & Bound, all solutions need to be enumerated in the worst case. This is basically done by traversing a search tree. In every node of the tree we divide the solution space into smaller subproblems, which are easier to solve. One way is to select a variable and in each subproblem or branch of the tree fix the variable to each of the values in its domain.

Another approach is to split the domain in two branches by adding the constraint $x \leq v$ to the left branch and $x \geq v+1$ to the right for $v \in D_x$, where D_x is the domain of the variable x .

For small problem instances as for instance 10 binary variables, we get 1024 potential solutions or leaves in the search tree. Increasing the number of variables to only 100 binary variables, we get a 31 digit number of potential solutions. This exponential growth makes it next to impossible to do an explicit enumeration of all solutions for problems of an interesting size.

Search by branching is the most obvious similarity between Branch & Bound and CLP.

Bounding by Relaxation

Branch & Bound is an implicit enumeration as it uses a bounding function to prune the search tree and only consider parts of the tree implicitly. A bounding function is found by relaxing some set of complicating constraints making the relaxed problem easy to solve. Let P be the region of solutions defined by the relaxed problem. Optimizing the objective function over P will give a lower bound z^{LB} on z^{opt} , since we are optimizing over a larger region.

Assume that the best solution found so far has value z^{best} . Now there are three cases where we can prune the search tree:

- $z^{LB} \geq z^{best}$ – no improving solutions in this subproblem.
- No feasible solutions can be found for the relaxed problem, and hence not for the original problem.
- A feasible solution is found to the original problem.

Bounding is not a part of the CLP paradigm. Instead optimization can be done in the following way: Each time a feasible solution has been found with objective value z^{best} a new constraint is imposed, $z < z^{best}$ making only improving solutions feasible.

The most widely used relaxation is *LP-relaxation* where the integrality constraints are relaxed resulting in a LP problem.

Strengthening by domain reduction and cuts

A very powerful extension of Branch & Bound is the possibility of adding *cuts* or constraints during the search to strengthen the relaxation. By finding and adding general or problem specific cuts, we want to remove parts of the relaxed solution space and ideally achieve the convex hull of the feasible

integer points. This procedure is referred to as *Branch & Cut*.

In Branch & Bound the tree can be pruned, if no feasible solution exists for the relaxed problem in the current node. In CLP infeasibility is detected when any variable gets an empty domain. The variable domains are reduced by use of *constraint propagation*. Consider the constraint $x_1 \geq x_2+1$ for $x_1, x_2 \in \{0, 1, 2\}$. No feasible value for x_1 exist when $x_2=2$. The value 2 can hence be removed from the domain of x_2 . Similarly is the value 0 in the domain of x_1 inconsistent with any remaining values of x_2 . At some stage in the search tree we would branch on x_1 and fix it to 2. Then the value 2 is not consistent in the domain of x_2 and it is removed and x_2 is fixed to 1. This is the most simple example of domain reduction, resulting in removal of the branching on x_2 .

The reduced domains of x_1 and x_2 are propagated to other constraints including x_1 and x_2 , which can again cause new domain reductions. This procedure continues until no further reductions can be performed and the resulting state is called consistent. Now 3 cases can occur:

1. All variables have exactly one value in its domain and a feasible solution is achieved.
2. One or more variables has no values in its domain – no feasible solutions.
3. At least one variable has a domain size larger than one.

In case of 1 and 2 the search backtracks or the branch is pruned in Branch & Bound terms. In case 3 a variable with domain size larger than one is selected and branching continues.

Generally a value can be removed from the domain of a variable, if the value is inconsistent with any combination of possible values of the other variables in the considered constraint. If all values in the domains of the variables are consistent, the constraint is called *consistent*.

There are different types of consistency:

Node consistency: The constraint only includes one variable.

Arc consistency: The constraint includes two variables.

Hyper-arc consistency: The constraint has more than 2 variables. In general it is *NP-hard* to show that a constraint is hyper-arc consistent and hence as hard as solving an arbitrary satisfiability problem [4].

In practice hyper-arc consistency is not used. Instead the concept of *bounds consistency* is introduced. Consider the

constraint $x = y+z$. Let $\min(D_x)$ and $\max(D_x)$ denote the minimum and maximum value in the domain of the variable x . The bounds on x can now be reduced by using the following constraints:

$$x \geq \min(D_y) + \min(D_z), \quad x \leq \max(D_y) + \max(D_z)$$

Isolating the other variables in the forms $y = x-z$ and $z = x-y$ we can reduce the domains of y and z in the same way - here shown for y :

$$y \geq \min(D_x) - \max(D_z), \quad y \leq \max(D_x) - \min(D_z)$$

Algorithms for reducing domains usually only consider one constraint at a time, but in many cases more reduction could be achieved by considering more than one constraint at a time. There is however a clear dependency between the complexity of the algorithm and the ability to reduce the domains. The choice of which algorithm to choose to get the best quality/time compromise can be problem dependent.

Domain reduction is not new in the OR community, since it has been a main part of specially tailored Branch & Bound algorithms. Domain reduction is as well used in presolvers of LP and MIP solvers, but only in the root of the search tree as the name also indicates. Note that from the point of view of Branch & Cut, domain reduction can be seen as adding simple cuts removing parts of the domain.

Designing Branch & Bound

In a combined framework of Branch & Bound and CLP the following design issues must be considered:

Bounding method: Choice of relaxation and bounding function.

Cuts and propagation: Choice of cuts to add as well as propagation strategies and algorithms.

Selection of next variable to branch on: Often you can do better than branch on a random variable. In binary branching pick a variable where fixing it to one value gives a low bound and to the other a high bound. Hopefully this will lead to a good feasible solution in the left subtree and when returning make it possible to prune the right branch. In CLP we generally want to branch on »important« variables first. For a given problem it is often quite obvious,

which variables are the most important. Later we will see an example on this. Generally branching on variables with a small domain size first is a good choice for minimizing the number of nodes in the tree.

Branching strategy: When subdividing the solution space, instead of branching on all values in the domain of a varia-

ble, which is common in CLP, the LP-relaxation can be used to split the domain in two. Let x' be the non-integral value of the LP-relaxation. The problem is then split into two subproblems given by the two constraints: $x \leq \lfloor x' \rfloor$ and $x \geq \lceil x' \rceil$.

The following branching example is often used when solving scheduling problems. Here the decision must be taken whether activity i should be executed before j or j before i . This can be embedded in the branching strategy. In one branch the constraint $t_i+p_i \leq t_j$ is added, where t is the start time and p the duration, and in the other branch the constraint $t_j+p_j \leq t_i$. This strategy is not available in a standard MIP solver.

The order in which values are tried in the usual CLP branching is often important. Perhaps it is known in advance that values in the middle of the domain are most likely to lead to good solutions. Incorporation of problem specific knowledge is necessary to make the search efficient.

Search strategy: Basically 3 pure strategies can be identified. In *Depth first search* a subtree is fully explored before backtracking, while in *Best first search* we always explore the most promising node first. Finally in *Breadth first search* all nodes in the same level of the tree are explored before continuing to the next level. The advantage of Depth first search is the limited memory consumption, since the number of active nodes will never be more than the maximum level of the tree. It is as well focused on finding a feasible solution as fast as possible in order to prune the tree. Breadth first search on the other hand will have as many active nodes as the level before the leaves. Best first search can in worst case be as bad as Breadth first, but generally it is much better. In fact after finding the optimal solution, no unnecessary nodes will be considered.

In CLP depth first search is used. When mentioning Branch & Bound and tree search, we implicitly think of *complete search* where all solutions are enumerated and optimum is guaranteed to be found. In some cases a »good« solution is sufficient and perhaps it is not possible to search the entire solution space within reasonable time. Then *incomplete search* can be useful. Here a number of different strategies are available to heuristically reduce the size of the tree. *Bounded Backtrack Search* simply stops the search after a certain number of backtracks has been exceeded. Other examples are *Credit Search*, *Limited Discrepancy Search* and others described in [1, 3], which more cleverly tries to explore promising regions.

Heuristics: To be able to prune the tree as soon as possible, you need a reasonably good feasible solution or upper bound before the Branch and Bound is started. Here a fast

construction heuristic can make the difference between solving the problem within seconds and not solving it at all. After finding an initial solution it can be improved by local search heuristics. Local search is well supported by ECLiPSe through the repair library, which allow variables to have tentative values [1, 3]. We will not discuss heuristics further in this paper.

$$\begin{aligned} \min \sum_{j \in J} c_j y_j + \sum_{i \in I} \sum_{j \in J} d_{ij} x_{ij} & \quad (5) \\ \sum_{j \in J} x_{ij} = 1, \quad i \in I & \quad (6) \\ x_{ij} \leq y_j, \quad i \in I, j \in J & \quad (7) \\ \sum_{i \in I} x_{ij} \leq cap_j, \quad j \in J & \quad (8) \\ y_j \in \{0,1\} \quad j \in J & \quad (9) \\ x_{ij} \in \{0,1\} \quad i \in I, j \in J & \quad (10) \end{aligned}$$

Figure 1: Warehouse Location Model

Modelling and Solving in ECLiPSe

ECLiPSe can use external LP and MIP solvers as CPLEX and XPRESS-MP and can hence be used purely as a mathematical modeling tool similar to GAMS. We will illustrate this by solving the following *Warehouse Location problem*: Given is a number of customers and a number of warehouse locations. Each open warehouse can only deliver to a limited number of customers and each customer only gets delivery from one warehouse. We must now decide which warehouses to open in order to minimize costs for opening the warehouses and delivering the demanded goods to the customers. The MIP model can be formulated as shown in figure 1 where c_j is the cost of opening warehouse $j \in J$ and d_{ij} is the cost of delivering goods to customer $i \in I$ from warehouse $j \in J$. The binary variable y_j indicates the opening of warehouse j while the binary variable x_{ij} indicates delivery to customer i from warehouse j . Constraint (6) guarantees delivery from exactly one warehouse to each customer. (7) forces delivery from only open warehouses. Finally (8) is the limit of customers serviced by each warehouse.

Introduction to ECLiPSe

We will for readers not familiar with Prolog or ECLiPSe give a short introduction to constructs used in the following.

- In Prolog the first letter in variables must be a capital letter. If a variable is assigned a value it is said to be *instantiated*. The only way a variable can be assigned another value is by backtracking past the place it was assigned the value and hereby becoming *uninstantiated*.
- A list is indicated by brackets: For $A = [X, Y, Z]$, A is

instantiated to the list $[X, Y, Z]$.

- $\text{dim}(X, [N, M])$ can be used for both constructing a matrix X or retrieving the dimensions of the matrix. If X is a given 2 dimensional matrix, the size is returned in the variables N and M . On the other hand if N and M are instantiated and X is not, then X will be initialized to a matrix with size N times M consisting of uninstantiated variables.
- $\text{length}(X, L)$ can be used for both constructing and retrieving the length, L , of a list X similar to dim for matrices.
- R is $X[I, 1..M]$ sets R to the I 'th row of the matrix X . Similarly C is $X[1..N, J]$ sets C to the J 'th column of the matrix X . R and C will be lists.
- $Y = 1, X = Y+1$ sets the variable X equal to the expression $Y+1$.
- $Y = 1, X$ is $Y+1$ sets the variable X equal to 2. Contrary to "=" is the expression on the right side evaluated when using *is*.

The last construct is the loop. We will give several examples to illustrate the different possibilities:

```
(for(I,1,10) do writeln(I))
```

Here the numbers from 1 to 10 are printed by the `writeln` statement.

```
(foreach(X, [1,2,3]) do writeln(X))
```

Prints the numbers 1, 2 and 3 by iterating over the list.

`foreach` can also be used for constructing lists:

```
(for(I,1,3), foreach(J,L) do J is I+1),  
writeln(L)
```

Note that it is not a nested loop, but simply two iterators.

Here the list $[2, 3, 4]$ is printed. Loops can also be used for constructing expressions:

```
(for(I,1,3), foreach(J,L) do J = I+1),  
writeln(L)
```

Here the list $[1+1, 2+1, 3+1]$ is printed. I and J are local variable in the loop while L is only accessible outside the loop. To make variables accessible within the loop `param` is used:

```
X is 3, (for(I,1,3), foreach(J,L), param(X)  
do J is I+X), writeln(L)
```

Here X instantiated to 3 is available inside the loop and the list $[4, 5, 6]$ is printed outside the loop.

MIP model in ECLiPSe

The ECLiPSe code of the Warehouse Location model is shown in figures 2 to 5. Calling the function

whouse (Cap, D, C, X, Y, Cost) will return the optimum value in the variable Cost. Cap is the vector of capacities, cap_j for warehouses $j \in J$, from the model in figure 1, D is the matrix, d_{ij} with distances from customer i to warehouse j and C is the vector of warehouse setup costs, c_j . Line 2 returns the dimensions of the matrix D, such that $NCust = |I|$ and $NWh = |J|$. NWh is used for initializing the vector, Y of y_j 's in line 3, and NCust and NWh are used for initializing the matrix X of x_{ij} 's in line 4. In line 5 `cross_prod(1..NCust, 1..NWh, Pairs)` returns the list Pairs of pairs [I, J] for all values of I and J in the intervals 1,...,NCust and 1,...,NWh to be used later.

```
1: whouse (Cap, D, C, X, Y, Cost) :-
2:   dim(D, [NCust, NWh]),
3:   dim(Y, [NWh]),
4:   dim(X, [NCust, NWh]),
5:   cross_prod(1..NCust, 1..NWh, Pairs),
```

Figure 2: Warehouse Location MIP model in ECLiPSe - Data.

In figure 3, the variable domains are defined. Line 6 creates a list YList of y_j variables, which are defined to be binary variables in line 7. When solving with an external solver we have to indicate that integer variables should be used with the call, `eplex: (integers(YList))`. Otherwise the LP-relaxation is solved instead, which can be useful in other circumstances. In line 9 a list XList of x_{ij} variables is extracted from the matrix X. In line 10 they are again defined to be binary. Here we do not need to tell the external solver to use integer variables, since the constraint matrix is unimodular for integral y_j 's.

```
6:   YList is Y[1..NWh],
7:   YList::0..1,
8:   eplex: integers(YList),
9:   matrix_to_list(X, XList),
10:  XList::0..1,
```

Figure 3: Warehouse Location MIP model in ECLiPSe - Variables.

In figure 4 we set up the constraints corresponding to (6), (7) and (8). First we loop over the customers where the variable I takes values from 1 to NCust. Line 14 constructs a list of x_{ij} variables for row i or here I. Line 15 sets up a constraint by summing the variables in the list and requiring this sum to be 1. Lines 16 to 18 iterates over all customer-warehouse pairs setting up constraint (7). Finally for each warehouse j or here J, row j is extracted from the matrix in line 21 and constraint (8) is set up in line 22.

```
11:  (for(I, 1, NCust),
12:   param(NWh, X)
13:   do
14:     XI is X[I, 1..NWh],
```

```
15:     eplex: (sum(XI) == 1)
16:     (foreach([I, J], Pairs),
17:      param(X, Y)
18:      do
19:        eplex: (X[I, J] =< Y[J])
20:      ),
21:     (for(J, 1, NWh),
22:      param(X, NCust, Cap)
23:      do
24:        XJ is X[1..NCust, J],
25:        eplex: (sum(XJ) =< Cap[J])
26:      ),
```

Figure 4: Warehouse Location MIP model in ECLiPSe - Constraints.

In lines 23 to 26 of figure 5 the customer part of the cost function is collected in the list CCost by iterating over all customer-warehouse pairs. Similarly are the warehouse setup costs collected in lines 27 to 29 and stored in the list WCosts. In line 30 the objective function is finally set up and in line 31 solved by calling the external solver with `optimize`.

```
23:  (foreach([I, J], Pairs),
24:   foreach(CCost, CCosts),
25:   param(D, X)
26:   do
27:     CCost = X[I, J] * D[I, J]
28:   ),
29:  (for(J, 1, NWh),
30:   foreach(WCost, WCosts),
31:   param(C, Y)
32:   do
33:     WCost = Y[J] * C[J]
34:   ),
35:  Objective = sum(WCosts) + sum(CCosts),
36:  eplex: optimize(min(Objective), Cost).
```

Figure 5: Warehouse Location MIP model in ECLiPSe - Objective.

It is clear that the above program is not as compact as the mathematical formulation or what could perhaps be formulated in another modelling language as for instance GAMS, but it is still relatively short considering it being written in a programming language. Actually what is taken the most space is the for loops.

CLP model in ECLiPSe

Now we set up the same problem as a CLP model shown in figures 6 to 9. Instead of using binary variables, y_j 's, we can think of choosing a set of warehouses to open. In line 3 we initialize a variable, Y, with `intset` to be a set with numbers between 1 and NWh.

```

1: whouse (CapList, D, C, XList, Y, Cost) :-
2:   dim(D, [NCust, NWh]),
3:   intset(Y, 1, NWh),
4:   length(XList, NCust),
5:   ( foreach(X, XList),
6:     param(Y)
7:     do
8:       X in Y
9:     ),

```

Figure 6: Warehouse Location CLP model in ECLiPSe - Variables

Instead of the binary variables, x_{ij} , specifying if customer i is serviced by warehouse j , we introduce variables for each customer taking values corresponding to open warehouses. The variables are initialized in the list `XList` in line 4 and in lines 5-7 they are constrained to take values `in` the set `Y` of open warehouses. We emphasize that `in` is a constraint, which could be mathematically stated like $x_i \in Y$, where x_i is the index of the chosen warehouse of customer i in the set-variable `Y` consisting of indices corresponding to open warehouses.

Note that instead of supplying the warehouse capacities in a vector, we now supply them in a list named `CapList`.

```

8:   ( for(J, 1, NWh),
9:     foreach(CapVar, CapVars),
10:    foreach(Cap, CapList),
11:    param(XList)
12:    do
13:      fd: (CapVar :: 0..Cap),
14:      occurrences(J, XList, CapVar)
15:    ),
16:   sum(CapVars) #= NCust,

```

Figure 7: Warehouse Location CLP model in ECLiPSe - Constraints

In lines 8-13 of figure 7 the capacity constraints are set up. We introduce the auxiliary variable `CapVar` in line 12 for each constraint. In line 13 we use the `occurrences` constraint, which constrains `CapVar` occurrences of warehouse `J` in the list `XList`. In line 14 we add the redundant constraint making the sum of `CapVar`'s equal to the total number of customers. As we will see later redundant constraints can have a significant impact on CPU times.

```

15:   ( for(I, 1, NCust),
16:     foreach(X, XList),
17:     foreach(CCost, CCosts),
18:     param(Y, D, NWh)
19:     do
20:       CostList is D[I, 1..NWh],
21:       element(X, CostList, CCost)
22:     ),

```

```

21:   weight(Y, C, WCosts),
22:   Cost #= WCosts + sum(CCosts),

```

Figure 8: Warehouse Location CLP model in ECLiPSe - Objective

To get the cost of supplying the customer from the chosen warehouse, we use the constraint `element` in line 20. `CostList` is the list of costs for the given customer and `element(X, CostList, CCost)` constrains `CCost` to take the X 'th element of the list `CostList`. The costs are collected in the list `CCosts` in line 17.

The costs for opening the warehouses, `WCosts`, defined by the set `Y`, is calculated with the `weight` constraint in line 21, where `C` is the vector of setup costs. Basically the weights corresponding to the set `Y` are summed in `WCosts`. For instance if the set is **{1,3,7}**, then the weights with index 1, 3 and 7 are added together.

```

23:   order_warehouses(D,
24:                     CustOrderedWarehouseIds),
25:   minimize((
26:     insetdomain(Y, increasing, _, _),
27:     labeling(XList,
28:              CustOrderedWarehouseIds)
29:   ), Cost).
30:   labeling(XList, CustOrderedWarehouseIds) :-
31:     ( foreach(X, XList),
32:       foreach(OrderedWarehouseIds,
33:              CustOrderedWarehouseIds)
34:       do
35:         member(X, OrderedWarehouseIds)
36:       ).

```

Figure 9: Warehouse Location CLP model in ECLiPSe - Search.

Remaining is now to specify the search procedure in figure 9. In line 27 `minimize` is called. It is a Branch & Bound procedure available in ECLiPSe. The procedure does not use a lower bound, which means that the performance of the search can be very much dependent on, in which order we consider variables and values. Therefore in line 23 we call a procedure, which for each customer sort the warehouses in increasing distance from the customer to warehouses. In line 26 we specify how to search the set of warehouses to open. `increasing` indicates that the cardinality of the sets we try are increasing, i.e. first we try sets with one element, then two, etc. In line 26 we call the search procedure shown in lines 28 to 32. Here the sorted list of warehouse id's are used to select values of `X`. `member(X, OrderedWarehouseIds)` basically iterates through the different values `X` can take.

Comparison of CLP and MIP

People from the CLP community would now argue that the CLP formulation is »closer« to how you would think of the real-life problem. OR people would argue instead that MIP models are just as easy to construct with a bit of modelling practice.

Another issue comparing CLP and MIP is performance. We have tested the two models on a small example with 19 warehouses and 20 customers for different warehouse capacities. We have as well tested different formulations where extra redundant constraints are added in order to improve performance. The following different models are tested:

MIP-S: Formulation in section 3.2.

MIP-LB: The following trivial lower bound on the number of open warehouses is added:

$$\sum_{j \in J} y_j \geq \left\lceil \frac{I}{\max_{j \in J} cap_j} \right\rceil \tag{11}$$

MIP-W: It is well known that

$$\sum_{i \in I} x_{ij} \leq cap_j y_j, \quad j \in J \tag{12}$$

is a weaker formulation than (7), but fewer constraints are necessary.

CLP-R: Formulation in section 3.3.

CLP-RLB: CLP-R with addition of same lower bound as for MIP-LB.

CLP-LB: As CLP-RLB without the redundant constraint in figure 7: `sum(CapVars) #= NCustomers`.

CLP: As CLP-R without the redundant constraint

The tests were done with ECLIPSe 5.5 and XPRESS-MP 13.26 and the results are shown in table 1. All warehouses have the the same setup cost and each column reports the results for a given capacity – the same for all warehouses. The necessary number of warehouses are then 2, 3, 4, 5, 7 and 10 for the given capacities.

The results of the aggregated and the stronger version of the MIP formulation are quite surprising. The weaker formulation is much faster on the difficult instances with many warehouses. The explanation is most likely that that the y 's are less fractional with many warehouses and the LP problem

easier to solve with much less constraints compared to the strong version. Adding the redundant lower bound on the number of warehouses improves the performance significantly.

Capacity	10	7	5	4	3	2
MIP-S	0.2	0.3	0.3	0.6	7.7	42.2
MIP-LB	0.2	0.2	0.1	0.1	0.1	0.4
MIP-W	0.2	0.1	0.1	0.1	0.2	0.1
CLP	4.5	318.2	-	-	-	-
CLP-R	4.7	8.8	19.3	50.0	251.3	1081.3
CLP-LB	4.5	7.9	14.8	34.0	126.8	566.8
CLP-RLB	4.7	7.9	15.2	35.0	132.9	585.8

Table 1: Comparison of CLP and MIP for the Warehouse Location Problem.

This is also the case for the CLP formulation. Here the lower bound is the difference making the problem solvable at all. Redundant constraints are often not necessary in MIP models, but for CLP they are. Experiments are as always needed to conclude if the constraints have any effect or if they are useless and even lead to increased runtimes. The other redundant constraint is not as powerful as the lower bound and the combination of the two is worse than the lower bound alone.

The reason the lower bound constraint is more effective, is the way the different sets of open warehouses are tried; first all sets with one warehouse open, then all sets with two open warehouses, etc. For each of the tried sets with too few open warehouses, customers are anyway allocated to warehouses, but without success.

We have earlier mentioned the importance of trying values in a specific order. In line 23 of figure 9 the warehouses are sorted in order of cost for each customer, which means that when trying to assign warehouses to customers it is done in increasing order of the cost. Without the ordering we get the runtimes shown in table 2, which are somewhat higher than CLP-LB.

Capacity	10	7	5	4	3	2
CLP-LB	4.8	8.4	25.14	43.0	158.8	839.3

Table 2: CLP-LB with no ordering.

Earlier we mentioned that branching should be done on important variables first. In this case the most important is, what warehouses to open. To illustrate the impact of making the right choice, we have tried the opposite order, i.e. decreasing sizes of sets to open, with the results shown in table 3.

Capacity	10	7	5	4	3	2
CLP-LB	130.5	456.7	-	-	-	-

Table 3: CLP-LB with branching on customer variables before warehouses.

The basis of the comparison is quite limited - only one quite small and simple prototype example. The intention of the comparison was however not to make the definite choice between MIP and CLP, but to merely illustrate some points.

On this instance of the problem MIP is superior to CLP, but it is risky business to generalize the result to other problems or even other instances of this problem. Problems for which there is a large gap between the upper and lower bounds are not well-suited for MIP. This for instance is the case when disjunctive constraints are necessary in scheduling and packing problems. Here CLP looks particularly promising.

Conclusion

We have in this article described CLP from the view point of MIP. The two paradigms have many similarities, but also the potential to complement each other and create even more powerful hybrid solvers.

We gave an example of modelling and solving the Warehouse Location Problem in the ECLiPSe environment with both MIP and CLP. We argued that MIP is superior when linear models can be formulated where the gap between upper and lower bounds is relatively small. Again CLP lacks the ability to use bounding to prune the tree, but can instead use domain reduction to reduce the size of the search tree.

Currently much interesting research is focused on using methods originating from the OR area to do domain reduction in CLP and as well using domain reduction in Branch & Bound. A definite conclusion is not yet near regarding, which components should enter into the future solver of NP-hard problems.

References

- [1] A. M. Cheadle, W. Harvey, A.J. Sadler, J. Schimpf, K. Shen, and M. G. Wallace. *ECLiPSe – An Introduction*. IC-Parc, Imperial College, London, August 2002.
- [2] W. F. Clocksin and C. S. Mellish. *Programming in Prolog*. Springer, 4. edition, 1994.
- [3] IC-Parc, Imperial College, London. *ECLiPSe Constraint Library Manual*, 2002.
- [4] K. Marriott and P. J. Stuckey. *Programming with Constraints – An Introduction*. MIT Press, 2. edition, 1998.



Af Jesper Hansen
(e-mail: jha@imm.dtu.dk)
Informatik og Matematisk Modellering
Danmarks Tekniske Universitet

Jesper Hansen er ph.d.-studerende og arbejder til dagligt med industriel anvendelse af matematiske modeller

Resources

We have collected some additional sources, if you are interested in getting more information on CLP and how to combine CLP with OR methods. Besides the OR newsgroup `sci.op-research` there is a constraints newsgroup `comp.constraints`, but it is not very active though there is a FAQ posted. More interesting is the »Constraints Archive« www.cs.unh.edu/ccc/archive, which has many links, but is perhaps not as updated as one could wish.

A large number of tutorials are available online, for instance the On-line Guide to Constraint Programming, by Roman Bartak at kti.ms.mff.cuni.cz/~bartak/constraints/. He has as well a tutorial on Prolog at kti.ms.mff.cuni.cz/~bartak/prolog/. If you prefer a more thorough introduction to constraints, I can recommend the book »Programming with Constraints - an introduction« by Marriott and Stuckey. It takes a more Computer Science approach than Mathematical Programming or OR approach. There has been written a large number of papers on combining CLP and OR in recent years, but I have not yet seen any tutorial type of papers introducing CLP from an OR perspective.

Download ECLiPSe from www.icparc.ic.ac.uk/eclipse and they have as well a lot of links to other related sites. People with OR background could as well be interested in attending the following conferences and workshops:

CP is the conference on the principles and practices of Constraint Programming.

CP-AI-OR is a workshop concerned with integration of AI and OR techniques in Constraint Programming.

PACLP is the short for the Practical Application of Constraint technologies and Logic Programming.

Civilingeniør **Brian Kallehaug** vinder DORS-prisen for 2000/2001

Efter en pause har DORS genoptaget traditionen med at præmiere et godt eksamensprojekt eller speciale. Hvert andet år uddeles DORS-prisen og ved selskabets julearrangement hos Nordea i december sidste år blev prisen for bedste eksamensprojekt/speciale afleveret i 2000 eller 2001 uddelt.

Specialerne indsendes af de respektive vejledere ved de videregående uddannelsessteder, og alle specialer er derfor af en sådan karakter, at de enkelte vejledere har ment, at de kunne vinde priskonkurrencen. Ud af de indsendte specialer må det konstateres, at det går godt for operationsanalysen i Danmark!

Der var denne gang indsendt fire hovedopgaver til bedømmelse og alle var karakteriseret af at være både gode og gennemarbejdede. De behandler hver især forskellige emner inden for operationsanalyse. De fire indsendte specialer er:

- »Lagrange dualitet og ikke-differentiabel optimering anvendt i rutelægning« af Brian Kallehaug (IMM, DTU)
- »Disruption management in the airline industri« af Michael Løve & Kim Riis Sørensen (IMM, DTU)
- »Placement of modules in VLSI layout« af Oluf Færø (DIKU, KU)
- »Multi-period portfolio management« af Jacob Lemming (IMM, DTU)

Kriterierne, som bedømmelsesudvalget i det store hele har gået ud fra har bl.a. været:

- Formålet med opgaven.
- Modtagers forudsætninger.
- Introduktion til området.
- Sammenfatning og konklusion.

Derudover, har bedømmelsesudvalget, idet dets medlemmer ikke er specialister udi ethvert af de indsendte specialer, set på hvordan specialerne overordnet set »virker«. Der er bl.a. lagt følgende betragtninger ned over de indsendte specialer:

- **Grundighed:** Er specialet grundigt gennemarbejdet? Er



Medlem af bedømmelsesudvalget Nina Dethlefsen (t.v) og DORS' kasserer Allan Larsen (mf) overrækker DORS-prisen til Brian Kallehaug (t.h.)

der en god blanding imellem den præsenterede teori og den anvendte analyse? Er der også en god blanding imellem den præsenterede teori og selve implementeringen, og finder implementationen gode resultater?

- **Soliditet:** Virker arbejdet i specialet solidt, er der gået tilbunds med de enkelte emner, som specialet behandler?
- **Originalitet:** Udemærker specialet sig ved på original vis at præsentere emnet og den tilhørende teori og endeligt, gives der originale bidrag til litteraturen i specialet?

Bedømmelsesudvalget mente, at Brian Kallehauges speciale »Lagrange dualitet og ikke-differentiabel optimering anvendt i rutelægning« udemærkede sig ved, meget overbevisende, at opfylde alle de opstillede kriterier. Brian Kallehauges speciale giver et meget sobert overblik over emnerne lagrange-dekomposition og ikke-differentiabel optimering og anviser metoder til løsning af rutelægningsproblemer - endda tidligere uløste, som oven i købet performer bedre end de hidtil kendte. Der er således tale om et originalt stykke arbejde, som også er kendetegnet ved at være både umådelig grundigt og solidt.

Bestyrelsen tiltrådte bedømmelsesudvalgets indstilling af Brian Kallehaug som derfor blev modtager af DORS-prisen. Bestyrelse benyttede samtidig muligheden til at takke bedømmelsesudvalget for et godt stykke arbejde. Bedømmelsesudvalget bestod af: Henrik Juel, IMM, DTU, Nina Dethlefsen, Danmarks Jordbrugsforskning, Tor Beltov (formand), SDU. [jla]