



Virksomhedsprojekt

Christian Mikkelsen Jensen, 060582
Anders Bjerg Pedersen, 070183

Analyse af
genindsættelse
af tog i DSB S-tog



Vejledere:
Pawel Winter (DIKU)
Jens Clausen (DSB S-tog)

Afleveret den 30. oktober 2009

Indhold

0.1. Indledning	2
0.2. Resumé	3
0.3. English abstract	3
1. DSB S-tog	4
1.1. Om virksomheden	4
1.2. Analysesektionen	5
1.2.1. Operationsanalytikerens	6
1.2.2. Konsulentens	7
1.2.3. Lederen	8
1.2.4. Samarbejdspartneren	8
1.3. Profilerings	9
1.4. Personlige refleksioner	9
1.4.1. Anders	9
1.4.2. Christian	11
2. Søjlegenerering	13
2.1. Søjlegenerering generelt	13
2.2. Dantzig-Wolfe dekomposition	14
2.3. Søjlegenereringsalgoritmen	15
2.4. Eksempel på søjlegenerering	17
3. Problem og modeller	21
3.1. S-togsnettet generelt	21
3.2. Krav og forstyrrelser	22
3.3. To mulige modeller	24
3.4. Model 1	26
3.5. Model 2	28
3.6. Sammenligning af modeller	30
4. Videreudvikling af modeller	33
4.1. Bløde krav	33
4.2. Alternative objektfunktioner	34
4.3. Ekstra disponible tog på depoterne	34
4.4. Fremtidig videreudvikling	35
5. Implementation og test	36
5.1. Implementation af modeller i GAMS	36
5.2. Testresultater	39
6. Konklusion	42
A. Testresultater for Linie C	44
B. GAMS-filer	45
C. Godkendt projektbeskrivelse	53

0.1. Indledning

Nærværende projekt tager udgangspunkt i en konkret problemstilling fra DSB S-togs afdeling for operationsanalyse. De fleste danskere kender sandsynligvis til S-togsnettet i København, men endnu flere kender måske til den situation, hvor ens tog er forsinket eller aflyst. Uforudsete hændelser, der skaber sådanne forsinkelser, vil uundgåeligt opstå på et eller andet tidspunkt, og derfor er det vigtigt for DSB S-tog at være bedst muligt forberedt på at håndtere dem. En af metoderne til at håndtere disse forsinkelser er at udtage hele toglinier på S-togsnettet. Dette projekt omhandler at genindsætte disse tog hurtigst muligt, efter forsinkelsen på den øvrige togdrift er afviklet. Umiddelbart virker dette problem simpelt, men på grund af et væsentligt antal krav i form af transport af lokomotivførere ud til togene, forskellige indsættelsesretninger (om et tog skal indsættes i nordgående eller sydgående retning), nogle mere togspecifikke krav samt hensynet til passagertilfredsheden, er dette problem langt fra trivielt.

Ovenstående situation opstår ofte i DSB S-tog. Man har før i tiden løst dette problem ved håndkraft og har derfor efterlyst værktøjer, der effektivt kunne håndtere og løse disse problemer i realtid. Analysesektionen hos DSB S-tog har udviklet to modeller til løsning af genindsættelsesproblemet, som hver især anvender meget forskellige metoder og til tider giver forskellige resultater. Dette projekt tager udgangspunkt i disse to modeller. For at udbrede perspektivet er projektet gjort til et virksomhedsprojekt, der skal klarlægge de overvejelser og metoder, der ligger bag relevansen, løsningen og arbejdet med problemstillinger af denne type i virksomheden.

Projektet er overordnet inddelt i 5 afsnit. Første afsnit omhandler virksomheden DSB S-tog, med særligt fokus på analysesektionen og dennes medarbejdere. Til sidst i afsnittet kommer vi med vores personlige refleksioner om virksomheden, dens arbejdsmetoder og -opgaver, samt projektets relevans for vores fremtidige studier og karriere.

Den ene af de to modeller fra DSB S-tog bygger på teorien bag søjlegenerering; en disciplin indenfor matematisk programmering. I afsnit 2 er temaet derfor søjlegenerering, hvor vi indledningsvist vil introducere Dantzig-Wolfe dekomposition, gå videre til selve søjlegenereringsalgoritmen og slutte af med et eksempel.

Afsnit 3 omhandler genindsættelsesproblemet. Først kommer vi med en introduktion til S-togsnettet, samt hvordan man i virksomheden håndterer forsinkelser. Derefter gennemgår vi de to modeller og sammenligner efter forskellige kriterier. I forlængelse af ovenstående kommer vi i afsnit 4 med konkrete bud på mulige udvidelser og videreudviklinger af de to modeller, ligesom vi til sidst i afsnittet foreslår mere langsigtede videreudviklinger. Afslutningsvist implementerer og tester vi ovenstående modeller og videreudviklinger i afsnit 5.

For at opnå fuld forståelse af det faglige indhold af projektet, forudsættes det, at læseren har gennemført et introducerende kursus i operationsanalyse, samt har kendskab til grundlæggende matematisk programmering, herunder specielt heltalsprogrammering. Afsnit 1, 2 og 3 kan læses uafhængigt af hinanden. Afsnit 4 og 5 bygger videre på foregående afsnit.

Vi vil gerne sende en stor tak til de ansatte i DSB S-tog, særligt til Jens Clausen og Birgitte Maribo Larsen for deres kyndige råd, vejledning og hjælp undervejs i projektet. Desuden vil vi gerne takke Pawel Winter for at finde tid til at vejlede os i dette projekt.

København, oktober 2009

Christian Mikkelsen Jensen
Anders Bjerg Pedersen

Forfattererklæring

Projektet er som udgangspunkt skrevet af os begge i fællesskab. Christian har skrevet afsnittene 1.4.2 og 5.2, og Anders har skrevet afsnittene 1.4.1 og 5.1. De resterende afsnit er følgende hovedansvarlig for:

Anders: 1.2.2-1.3, 2.1, 2.2, 3.2, 3.3, 3.6, 4.1 og 4.3.

Christian: 1.1-1.2.1, 2.3, 2.4, 3.1, 3.4, 3.5, 4.2 og 4.4.

0.2. Resumé

Dette virksomhedsprojekt omhandler DSB S-tog og genindsættelsesproblemet, der vedrører genindsættelse af udtagne toglinier. Projektet indledes med en grundig beskrivelse af virksomheden, specielt analysesektionen og dennes medarbejdere, for derefter at opstille de nødvendige teoretiske værktøjer og metoder i form af Dantzig-Wolfe dekomposition og søjlegenerering. Vi gennemgår to forskellige modeller, som er udviklet af DSB S-tog, sammenligner disse og kommer blandt andet frem til, at en af modellerne ikke opfylder et ellers hårdt stillet krav. Afslutningsvist ser vi på mulige videreudviklinger af de to modeller, og til sidst implementerer og tester vi dem op mod hinanden.

0.3. English abstract

This enterprise project deals with the Danish railway operator DSB S-tog and the train reinsertion problem, which consists of reinserting cancelled train lines. Initially we thoroughly describe DSB S-tog as a company, with special attention put on the analysis section and its staff members. After this we set up the required theoretical tools and methods in the form of Dantzig-Wolfe decomposition and column generation. We examine two different models that have been developed by DSB S-tog, compare these two models and – among other observations – discover that one of the models does not fulfil an otherwise explicitly stated hard constraint. Finally we review possible further developments of the two models and ultimately implement and test them opposite each other.

1. DSB S-tog

Første del af projektet kommer til at handle om DSB S-tog, og specielt analysesektionen, som vores projekt er skrevet for og hos. Vi starter med et overblik over virksomheden og fortæller kort om dens historie og relation til modervirksomheden DSB. Desuden skal vi se på de arbejdsopgaver, der ligger hos S-tog generelt og mere specifikt i analysesektionen, medarbejderne og disses kompetencer og baggrund samt de arbejdsformer, der bliver anvendt i virksomheden. Herudover vil vi komme ind på samarbejdsfladerne med andre sektioner i virksomheden, sektionens profilering (faglig og social) samt sektionens relation til forskningsmiljøerne.

Informationer og betragtninger i denne del bygger mestendels på interviews med ansatte i analysesektionen hos DSB S-tog, samt egne refleksioner og forventninger til en kommende arbejdsplads. Interviews er gennemført med 5 personer fra virksomheden: sektionslederen for analysesektionen, to operationsanalytikere, en deltidsansat konsulent og en personaleplanlægger fra produktionsplanlægningssektionen.

1.1. Om virksomheden

DSB S-tog er et datterselskab af DSB, som er en fuldt statsligt ejet virksomhed. DSB S-tog A/S blev i 1999 udskilt som aktieselskab, der dog er 100% ejet af moderselskabet DSB. De første S-tog kørte dog over skinnerne helt tilbage i 1934, og "S"et i S-tog kom man frem til gennem en rundspørge i Politiken i 1934, sandsynligvis fordi man allerede havde opsat de karakteristiske skilte for at markere, at der lå en station.

Da S-tog er et aktieselskab, har det selvfølgelig en bestyrelse. Denne består af 6 personer, blandt dem DSBs administrerende direktør, som er formand. Internt i DSB S-tog er virksomheden opdelt i 4 afdelinger med hver deres ansvarsområder. Denne organisation kan ses i Figur 1.



Figur 1: Den interne organisation i DSB S-tog

DSB S-tog varetager under kontrakt med Transportministeriet driften af S-togsnettet i København og omegn. Dette indebærer den daglige drift af togene (lokomotivførere, togrevisorer, køreplaner, passagerinformation, etc.), rengøring og vedligeholdelse af tog og stationer på nettet, sikkerhed samt kvalitetssikring af driften. S-tog varetager *ikke* selv vedligeholdelse og udbygning af skinnenet og signalsystemer. Denne opgave varetages af BaneDanmark. Dagligt bruger omkring 300.000 mennesker S-togsnettets 7 linier med tilhørende 84 stationer. Dermed udgør passagertallet for DSB S-tog den betydeligt største del af det samlede passagertal for hele DSB-koncernen. ([1], [4] s.2ff)

1.2. Analysesektionen

Analysesektionen (AS) er en underafdeling af produktionsplanlægning, som igen er en underafdeling af Produktion (se Figur 1). Produktionsplanlægning består af 4 sektioner: analyse, tjenesteplanlægning (personale), materielplanlægning og en projektgruppe. De første operationsanalytikere kom til DSB S-tog omkring år 2000 som et svar på en efterspørgsel efter større og bedre integration mellem de underliggende datasystemer, der var under anvendelse i virksomheden, ligesom der var et stort behov for at mindske de u hensigtsmæssigheder, der var med de daværende køreplaner og daglige udfordringer. AS skulle dermed, via udpræget brug af blandt andet matematisk modellering og IT-systemer, hjælpe til med at optimere og udvikle driften af S-togsnettet, samt skabe større relation mellem køreplaner og ”den virkelige verden”. Selve AS blev først formelt oprettet som selvstændig sektion omkring 2005. Desuden er AS med til at forberede DSB S-tog på kvalificeret at kunne byde ind på en eventuelt fremtidig udlicitering af S-togsdriften. AS leverer ofte input til andres beslutninger, da de har et indgående kendskab til virksomheden. Denne indsigt er dog begrænset, og politiske spillere som f.eks. fagforeninger kan stoppe konkrete projekter. En pendant til analysesektionen findes ikke i DSB-koncernen på overordnet plan. Derfor bliver AS involveret i udbudsrunder, som DSB kunne tænkes at byde ind på. Disse opgaver prioriteres særdeles højt.

Analyseafdelingen består af 6 medarbejdere: en sektionsleder, fire fuldtidsmedarbejdere og en deltidsmedarbejder. Alle medarbejdere har akademisk baggrund, og hovedparten er ingeniøruddannede, mange med operationsanalyse som speciale. Som den eneste uden ingeniørbaggrund er konsulenten uddannet cand.scient i datalogi og er ansat deltid hos S-tog og deltid på DTU. Denne sammensætning harmonerer fint med sektionens målsætning om at holde en akademisk indgangsvinkel til arbejdsopgaverne, såvel som at være bindeled mellem den overordnede akademiske tilgang til opgaver og de konkrete praktiske problemstillinger. Dette kommer endnu tydeligere til udtryk i samarbejdet med andre afdelinger og sektioner (mere herom senere i dette afsnit).

De mere konkrete arbejdsopgaver i sektionen omfatter blandt andet udvikling og vedligeholdelse af IT-systemer til for eksempel produktionsplanlægning (mandskabs- og køreplaner, materieldisponeringssystemer o.lign.), indsamling, konvertering, vedligeholdelse og bearbejdning af data fra for eksempel passagertællinger og statistik. Overordnet ligger fokus på brug og udvikling af operationsanalytiske metoder og værktøjer. Sektionen samarbejder i denne forbindelse med mange andre sektioner og afdelinger, men primært internt i produktionsplanlægning. Samarbejdet foregår hovedsageligt i planlægnings- og implementationsfasen; AS trækker sig mere ud af opgaven, når samarbejdspartneren er ved at kunne overtage driften.

Arbejdsformen i AS er primært projektorienteret. Sektionslederen er ansvarlig for udvælgelse og fordeling af projekterne. Hvert projekt tildeles en projektansvarlig samt typisk yderligere 1-2 projektdeltagere. Det tilstræbes, at der altid er mindst én sparingspartner på hvert projekt. Projekterne er af forskellig varighed, og involverer andre sektioner og afdelinger, alt efter projekttype og omfang. Der foregår en del videndeling i sektionen på forskellige måder. Først og fremmest foregår en meget stor del af denne videndeling på ugentlige sektionsmøder. Her kommer medarbejderne med status på deres projekter, en "vejmelding" på hvad de konkret sidder og arbejder med, samt af og til fremlæggelser af deres projektræsultater. Desuden arrangeres der internt i sektionen af og til små mini-kurser for at genopfriske eller introducere ny teori eller teknikker. Én gang om året tager sektionen på ekskursion, også kaldet "strategidage". Her diskuteres det, hvordan sektionen i det kommende år (eller på længere sigt) kan prioritere deres ressourcer bedst muligt for at generere mest mulig værdi for virksomheden. Ligeledes gennemføres SWOT-analyser¹, og der afholdes et årligt personaleseminar, hvor det sociale er i højsædet. Her diskuteres også mere medarbejderrelaterede emner som efteruddannelse (f.eks. inden for formidling), deltagelse i konferencer og lignende.

For at illustrere, hvad der konkret arbejdes med i AS, vil vi nu på baggrund af ovennævnte interviews kort beskrive nogle af sektionens medarbejdere, deres opgaver og baggrund. Desuden har vi interviewet en medarbejder fra en anden sektion, der har stor samarbejdsflade med AS.

1.2.1. Operationsanalytikeren

Vi har i denne kategori talt med to ansatte i sektionen, der for det første har operationsanalytisk baggrund, og som for det andet har området som deres primære arbejdsfelt. Begge er uddannet civilingeniører, den ene med erhvervsmæssig baggrund i Forsvaret indenfor forskning og undervisning i operationsanalyse, den anden med en kortere konsulentbaggrund forud for ansættelsen hos DSB S-tog. Begge kom ind i virksomheden gennem en form for personlig kontakthade med virksomheden; den ene gennem konsulentarbejde udført for virksomheden for 10 år siden, den anden gennem personligt netværk til sektionen fra uddannelsesstedet.

De to operationsanalytikere sidder med i en del forskellige arbejdsopgaver og projekter i virksomheden. Blandt andet inden for omløbsplanlægning, der beskæftiger sig med, hvordan materiellet skal fordeles på samtlige af S-togs strækninger. Hvor skal toge stå om aftenen, skal noget flyttes rundt om morgenen, hvilken sammensætning (længde) skal de forskellige tog have (mere om de forskellige togtyper på side 22). I forbindelse hermed er de begge med på et projekt, der skal implementere et nyt system (OMPLS: OmløbsPlanlægningsSystem), som kan varetage og forbedre den daglige materielplanlægning. OMPLS bygger på matematiske modeller, og er baseret på svensk forarbejde.

Den ene af operationsanalytikerne deltager desuden i projektet APGS: Automatisk PassagertalsGenereringsSystem. S-tog er i fuld gang med at overgå fra sædvanlige passagertællinger til udelukkende at udnytte data fra vejning af hvert tog for på denne måde

¹SWOT: Strengths, Weaknesses, Opportunities, Threats. Et strategisk værktøj, der kan hjælpe virksomheder til at prioritere ressourcer overfor udfordringer (se f.eks. <http://www.quickmba.com/strategy/swot/>).

at tælle antallet af passagerer i toget. APGS har til formål at udvikle et system, der kan analysere alle disse eksisterende og nye data på en lettilgængelig måde. Det er på dette område, AS skaber størst gevinst for virksomheden.

Begge anvender langt de fleste af deres faglige kompetencer fra uddannelse og tidligere erhvervserfaring direkte i deres arbejde hos DSB S-tog. Fokus ligger altså på de operationsanalytiske kompetencer samt en del statistik. En af medarbejderne udtalte dog et behov for flere kompetencer inden for formidling. Dette er nødvendigt specielt i samarbejdet med andre afdelinger med anden faglig baggrund, og vedkommendes uddannelse har ikke indeholdt sådanne elementer.

Specielt på OMPLS-projektet er der en stor kontaktflade til andre sektioner i produktionsplanlægningen og i selve DSB, hvor der findes en lignende afdeling, der varetager omløbsplanlægning. Efter udviklingen af et nyt system eller en model er der til tider en vis modstand fra andre faggrupper, når det kommer til selve implementationen i den daglige rutine. Her ligger der en stor opgave for analysesektionen i at formidle deres resultater på en måde, så det giver mening for andre faggrupper at ændre deres arbejdsvaner for at få implementeret resultaterne i den daglige drift. Herudover er der en del sparring og samarbejde med andre relevante afdelinger om diverse projekter og arbejdsopgaver, når der er behov for dette.

Begge medarbejdere er enige om, at det er en attraktiv arbejdsplads, hvor man får lov til at sætte sit præg på passagerernes rejseoplevelse og overordnede indtryk af virksomheden. Der er masser af udfordringer og god kemi, og der er god mulighed for at deltage på diverse faglige seminarer og møder, både i det faglige miljø omkring virksomheden, samt i det akademiske miljø. Begge regner med at blive i virksomheden i lang tid endnu.

1.2.2. Konsulenten

Konsulentens primære funktion er at skabe et bindeled og kontaktflade mellem det "tog-faglige" miljø og det akademiske miljø. Dette opnås på forskellige måder. Konsulenten fungerer blandt andet som peer-reviewer på videnskabelige artikler, der har med produktionsplanlægning og togdrift at gøre. Hermed er han med til at videreføre viden til virksomheden, allerede inden den er publiceret, samt er medvirkende til at holde de andre medarbejdere opdateret og forankret i det akademiske miljø. Han har desuden mulighed for at gå mere teoretisk i detaljer med konkrete projekter i virksomheden, som de fastansatte medarbejdere ellers ikke ville have overskud eller tid til. Derudover har konsulenten en stor teknisk indsigt i matematiske modeller og de resultater, som kan forventes af dem.

De faglige kompetencer, som konsulenten er i besiddelse af, bliver mestendels brugt i form af matematisk-tekniske kompetencer. Han er desuden i besiddelse af en del formidlingskompetencer, som han kun i mindre omfang finder anvendelse for i virksomheden. Kontaktfladen og samarbejdet med andre afdelinger og personer er grundet tilknytningen til mange forskellige projekter meget bred, men typisk knap så dyb.

1.2.3. Lederen

Lederen af AS kommer ligeledes fra en operationsanalytisk baggrund med en ingeniøruddannelse i informatik. Hun reagerede i første omgang på et stillingsopslag under specialeskrivningen i 2003, hvor de efterspurgte kompetencer matchede hendes egne rigtig godt. I starten var der kun to ansatte, der refererede direkte til chefen for produktionsplanlægning. Efterhånden som de ansatte viste gode og brugbare resultater, kom der flere ansatte til, og til sidst var der berettigelse for at oprette en sektion for sig selv. Efter afsluttet Ph.d overtog hun posten som sektionsleder, da den tidligere forlod virksomheden.

Sektionslederens arbejdsopgaver adskiller sig væsentligt fra de øvrige medarbejderes, idet der her er tilføjet hele ledelsesaspektet. Hun er i øjeblikket kun med som sparringspartner på et enkelt projekt omhandlende mandskabsplanlægning i realtid og regner med, at dette bliver det sidste konkrete projekt indtil videre. De sædvanlige ledelsesmæssige opgaver (såsom fordeling og prioritering af projekter, ledelsesinformation op i systemet, etc.) skal selvfølgelig udføres. Med hensyn til udvælgelse af projekter foretrækker lederen, at projekter kommer eksternt til sektionen. Dette gør det nemmere for AS at forsvare vigtigheden af de projekter, de laver, samt letter implementeringen af resultaterne i de sektioner, som har stillet projektet. Internt forsøger lederen at fordele projekter efter kompetencer og interesser.

Sektionslederen anvender dagligt sine operationsanalytiske kompetencer til at forholde sig til konkrete problemstillinger. I sin egenskab af leder er de dog ikke praktisk så meget i spil som andre medarbejdere med samme kompetencer. Hun er dog også i besiddelse af en grundlæggende lederuddannelse, som bruges i det daglige arbejde med at være lydhør, motivere medarbejdere, stole på deres viden og drive dem fremad.

1.2.4. Samarbejdspartneren

Denne medarbejder sidder i den sektion, der har med mandskabsplanlægning at gøre. Her er der specifikt tale om vagtplanlægning for den enkelte lokomotivfører. Han startede som lokomotivfører i DSB, men senere kom han med som medhjælper på forskellige projekter i planlægningsafdelingen hos en af dennes faste medarbejdere. Da denne medarbejders stilling blev slået op, søgte vedkommende den og blev efterfølgende fastansat i planlægningsafdelingen. Dette er ikke nogen usædvanlig karrierevej, da alle ansatte i sektionen har baggrund som lokomotivførere.

De specifikke arbejdsopgaver indebærer for tiden sammenligning og udvikling af to mulige systemer til mandskabsplanlægning, ligesom de mere faste opgaver består af den konkrete planlægning på lang og kort sigt. Meget af samarbejdet med AS vedrører for tiden netop denne sammenligning af systemer, hvor analysesektionen kan være med til at kvalitetssikre og videreudvikle de to systemer overfor hinanden. Herunder at afgøre, om systemerne hver især løser de opgaver, som virksomheden gerne vil have dem til, og på den bedste måde. AS fungerer her som en slags problemknusere eller eksperter på de områder, der har med "indmaden" i systemerne at gøre. Disse to systemer bygger ligeledes på intensivt brug af matematiske modeller og løsningsmetoder.

AS har i deres arbejde også brug for informationer den anden vej, i form af for eksempel spadseretider fra perron til personaledepot, pauser og overenskomstmæssige krav. Grunden hertil er blandt andet, at togplanlæggerne ofte ligger inde med en del "tavs viden", altså viden der er nødvendig for at udføre planerne i praksis, men som ikke er nedskrevet nogen steder. Da samarbejdet foregår mellem to meget forskellige typer af faggrupper, opstår de største komplikationer i forhold til samarbejdet i selve kommunikationen og tilgangen til opgaverne. AS er meget detaljefokuseret, hvor togplanlægning er mere praktisk og løsningsfokuseret på kort sigt.

1.3. Profilerings

Analysesektionen har ikke som sådan en selvstændig profil, de gerne vil gøre sig kendt på. De forholder sig mestendels til de overordnede strategier, som DSB S-tog og DSB har udstukket. Sektionen vil selvfølgelig helst fremstå som en kompetent arbejdsplads, og den store interne sparring gør virksomheden særligt interessant for folk med operationsanalytisk baggrund. Derfor søger sektionen ansatte med forskellige kompetencer inden for operationsanalyse, statistik og IT, i prioriteret rækkefølge. Erhvervserfaring er ikke nødvendigvis et krav, hvis ansøgeren kan udvise evne til at kunne føre operationsanalytiske koncepter og metoder ud i praksis. Desuden er det en fordel, at man kan formidle sin faglighed; altså ligger inde med en vis form for diplomatiske og formidlingsmæssige evner. Da antallet af medarbejdere er meget begrænset, vil man dog i en ansættelsessituation se meget på, hvem der blandt de eventuelle ansøgere passer bedst ind i teamet.

Da AS er en relativt ny sektion, er det svært at sige noget om, hvor attraktiv arbejdspladsen er, målt på hvor længe medarbejderne vælger at blive der. De fleste af de nuværende medarbejdere har været der i over 3 år, og vi har indtryk af, at det generelt ikke er en sektion, man bruger som springbræt til andre eller "bedre" jobs, på trods af at lønniveauet er godt uden at være prangende.

For at fastholde medarbejdere i afdelingen og sikre kontakten til de akademiske miljø, kan medarbejdere deltage i 1-2 konferencer om året, såfremt det er relevant for deres arbejdsopgaver i AS. Herudover afholdes der hvert år en del faglige og sociale arrangementer i DSB S-tog, som er med til at sammentømre medarbejdere og sektioner. De faglige arrangementer kan blandt andet bestå af møder og sparring med andre eksterne virksomheder og samarbejdspartnere. De sociale arrangementer foregår mestendels sammen med hele DSB S-tog og være sig julefrokoster, DHL-stafet og flere andre.

1.4. Personlige refleksioner

I det følgende afsnit vil vi hver især kort beskrive vores motivation for at skrive et virksomhedsprojekt. Fokus ligger derudover på vores refleksioner over det at arbejde med operationsanalyse i en virksomhed, samt hvordan disse erfaringer kan bruges i vores fremtidige uddannelse og senere arbejdsliv.

1.4.1. Anders

Mine første indskydelser, når jeg tænkte på ordet "S-tog", var ord som "forsinkelser", "signalproblemer", "DSB" og "komplekst". Da chancen viste sig for, at jeg kunne komme til at skrive et virksomhedsprojekt hos DSB S-tog, så jeg det som en oplagt mulighed for

at få sat lidt flere ord på denne ukendte størrelse, samtidig med at jeg kunne få prøvet mine faglige kompetencer af uden for universitetsmiljøet. Det har dog også gennem projektet vist sig, at der skulle en del flere kompetencer i brug end blot de faglige.

Indledningsvist var der problemer med overhovedet at finde en virksomhed, da der på fakultetet endnu ikke er en velopbygget vidensdatabase for virksomhedsprojekter og de mulige samarbejdspartnere. Vi havde senere heller ikke mulighed for at se eksempler på projekter eller lignende, så i starten var det uhyre svært at spore sig ind på, hvad vi havde mulighed for at beskæftige os med i projektet. At vi endelig fandt frem til DSB S-tog, havde i høj grad noget med vores faglige netværk at gøre. De forskellige kurser, som jeg har været igennem på datalogistudiet, har selvfølgelig medført, at jeg kender til en del undervisere og disses arbejdsområder. At gå direkte til dem viste sig at være en god plan, da vi så gennem det akademiske miljø fik kontakt videre ud til erhvervslivet. På dette punkt har vi nok været heldige, da vores vejleder i virksomheden er gammel ansat på datalogisk institut. Processen omkring rent faktisk at finde en virksomhed og et projekt har altså krævet meget stor selvstændighed, udadvendthed og initiativ fra vores side, men i sidste ende har det bestemt været indsatsen værd.

Som person er jeg selv, ligesom Christian, ikke bange for at skabe nye kontakter og efterfølgende også udnytte dem. Disse kompetencer har vi haft brug for, specielt i begyndelsen af projektet. Desuden har jeg i min egenskab af studievejleder den fornødne tekniske indsigt til at få den overraskende store mængde formalia omkring projektets gennemførsel til at gå op.

Med hensyn til anvendelsen af mine faglige kompetencer, så har dette projekt været lidt af en øjenåbner for mig. Jeg har længe forsøgt at finde ud af, hvad "sådan én som mig" kan bruges til ude i erhvervslivet. Igennem uddannelsesforløbet er der ikke de store muligheder eller overskud til at relatere det faglige til eventuelle senere ansættelsesmuligheder. Her kommer virksomhedsprojektet ind som løsningen. Dette projekt har vist mig et glimrende eksempel på, hvad lige præcis en person med mine kompetencer vil kunne komme til at lave efter endt uddannelse. Specielt gennem de interviews, som vi har gennemført med en håndfuld ansatte i analysesektionen, har jeg fået øjnene op for, at disse ansatte i høj grad arbejder med netop de kompetencer, som jeg er ved at opbygge: indsigt i matematisk modellering og operationsanalyse, solid matematisk baggrund, programmeringserfaring og projektarbejdskompetencer. Matematisk modellering og operationsanalyse har jeg oparbejdet kompetencer indenfor blandt andet gennem kurserne "Operationsanalyse 1 (OR1)", "Videregående Algoritmik (VA)", "Optimeringsproblemer i Produktionsplanlægning (OPPP)" samt et kandidatprojekt om tabu-søgning, men specielt i vores fælles bachelorprojekt "Sports Scheduling". Den solide matematiske baggrund ligger selvfølgelig i min bachelorgrad i matematik, mens kombinationen med datalogi som tilvalg og en efterfølgende kandidatuddannelse i samme har givet de resterende kompetencer.

Jeg er vant til projektarbejde fra mit nuværende studiejob, så det var positivt for mig at se, at denne arbejdsform også fandt meget stor anvendelse hos DSB S-tog. Arbejdsformen lader til at være den drivende faktor i opgaveløsningen i virksomheden, og dette tiltaler mig meget. Ligeledes er der en forholdsvis bred samarbejdesflade til andre afdelinger og faggrupper, som kræver en vis grad af formidlingskompetencer. Disse er

jeg også i besiddelse af gennem begge mine studiejobs, som har et væsentligt indhold af formidling: evnen til at kunne formidle faglig viden fra én gruppe eller vidensniveau til et andet. Det sociale miljø virker også godt sammentømret; en faktor, der spiller en ganske væsentlig rolle i mit fremtidige valg af arbejdsplads. Disse ting, kombineret med at projekterne fordeles efter interesse og kompetencer, er med til at gøre DSB S-tog til en meget attraktiv arbejdsplads for mig.

Det har været meget inspirerende at arbejde hos og sammen med DSB S-tog. Jeg har helt sikkert fået et solidt indblik i, hvad det vil sige at være matematikinteresseret datalog i det private erhvervsliv. Hvorvidt det er det samme område, jeg skal fortsætte mine studier i, ved jeg ikke, men det har været betryggende at finde ud af, at den vej i hvert fald ikke fører til uinteressante jobs. Så et speciale inden for operationsanalyse ligger nok ikke så langt ude i fremtiden.

1.4.2. Christian

Anders og jeg har, lige siden vi skrev bachelorprojekt sammen, overvejet muligheden at skrive et projekt sammen igen. Vi var også begge meget interesserede i at prøve at skrive et virksomhedsprojekt, da vi så for alvor ville få en forståelse af, hvordan vores fag fungerede i en erhvervsmæssig sammenhæng. Projektet har da også givet os mange udfordringer, ikke kun på det rent faglige men også det mere personlige plan. At skrive et virksomhedsprojekt kræver i sagens natur en virksomhed, hvilket har krævet en del personlige kompetencer i den indledende proces med blandt andet at tage initiativ til at kontakte virksomheder. Herefter skulle vi have projektet godkendt, samt få dispensation til at skrive sammen, da vi læser hver vores overbygning. Her udnyttede vi hinandens personlige kompetencer, Anders er studievejleder og dermed godt inde i regler med f.eks. dispensationer, mens jeg stod for den indledende kontakt til DSB S-tog. Disse opgaver var ikke nye for mig, da jeg dels tidligere i mit studie har kontaktet eksterne personer til vejledning og ekspertise, og dels gennem frivilligt arbejde og mit studiejob jævnligt kontakter og møder nye mennesker.

Selve projektet lå i fin forlængelse af de kurser jeg har haft tidligere på mit studium. Specielt i forhold til et kursus (Videregående Operationsanalyse), som jeg fulgte sideløbende med projektet i foråret 2009. Kursets indhold var primært heltalsprogrammering, men berørte også kort søjlegenerering og vi opstillede nogle mindre modeller i GAMS. Derudover har jeg brugt en del faglige kompetencer som er erhvervet igennem hele uddannelsen, det være sig matematisk kommunikationskompetence, modelleringskompetence og tankegangskompetence. En del af projektet omhandlede selve DSB S-tog, hvilket igen gjorde brug af en del sociale kompetencer.

Selve analysesektionen og måden der arbejdes på, har overrasket mig en del. Det, der nok har overrasket mig mest, har på den ene side været den tætte tilknytning til den akademiske verden, men på den anden side hvor lidt der i nogle tilfælde var at finde på skrift. Det sidste bygger jeg på model 2 (se afsnit 3.5), hvor der slet ikke var noget på skrift. Arbejdsformen havde jeg inden ikke tænkt så meget over, men jeg ville bestemt godt kunne se mig selv arbejde i en virksomhed med en meget projektorienteret indgangsvinkel, som det er tilfældet med DSB S-tog. Noget af det, som jeg finder meget attraktivt i DSB S-tog, er den store kontaktflade med mange andre afdelinger og sektioner. Det at

kunne formidle et budskab til personer med en helt anden faglig baggrund end en selv, synes jeg er rigtig spændende, hvilket nok også er grunden til, at jeg har beskæftiget mig meget med netop dette privat. En anden faktor, der virker meget tiltalende, er den store tillid der er i sektionen. Det at man kan arbejde hjemme, tage på konference med mere.

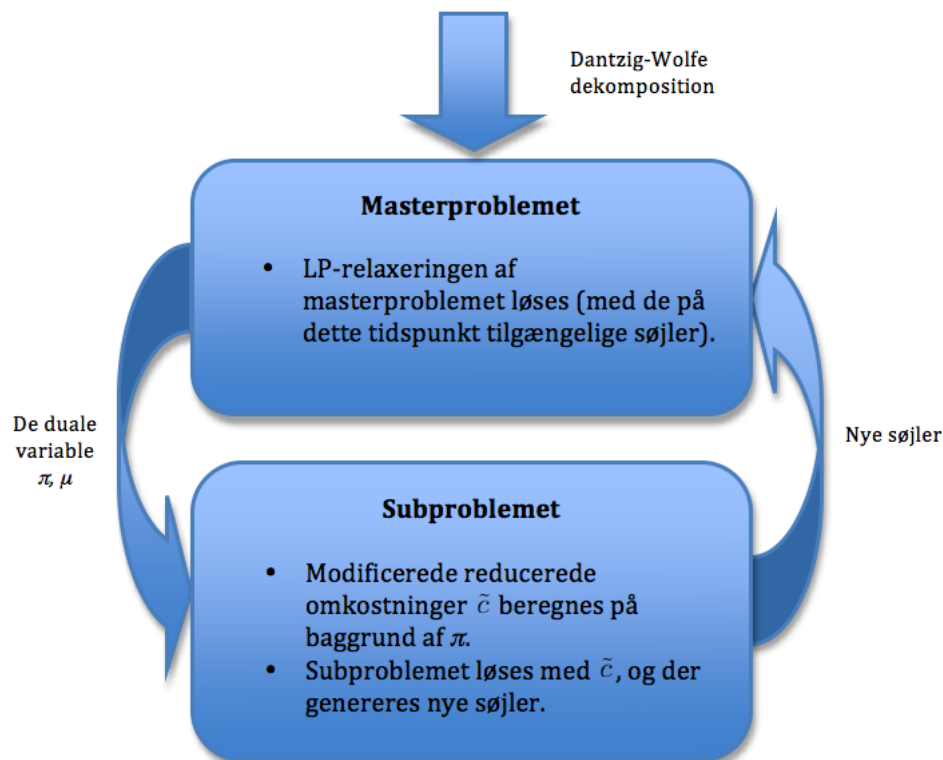
Det at komme til at arbejde med helt konkrete problemstillinger i en virksomhed, har givet en rigtig god indsigt i, hvordan operationsanalyse kan bruges i erhvervslivet. Omtrent et halvt år før projektet startede, havde jeg store overvejelser omkring, hvordan jeg skulle sammensætte den sidste del af min uddannelse, om det skulle være mod operationsanalyse eller mere mod det didaktiske og gymnasielærer. Her har dette projekt i høj grad bekræftet mig i, at det var rigtigt at vælge operationsanalyse.

2. Søjlegenerering

I det følgende afsnit skal vi se på en problemløsningsmetode til løsning af heltalsproblemer. Vi skal senere se, hvordan en af de to modeller, vi præsenterer til løsning af genindsættelsesproblemet, bygger på denne fremgangsmåde. Afsnittet indledes med en motivation for overhovedet at beskæftige sig med søjlegenerering, derefter fortsætter vi med at identificere de problemstillinger, hvor det finder anvendelse, og til sidst gennemgår vi teorien og fremgangsmåden bag selve metoden. Afslutningsvis vil vi som eksempel anvende teorien på det velkendte *Symmetric Travelling Salesman Problem (STSP)*.

2.1. Søjlegenerering generelt

Teorien bag søjlegenerering er forholdsvis ny. Den slog først rigtigt igennem i 1980'erne, blandt andet hos J. Desrosiers, F. Soumis og M. Desrochers. Det overordnede princip med søjlegenerering er at tage ens oprindelige problem og spalte op i to delproblemer: et masterproblem og et subproblem. Denne opdeling laves på baggrund af, at det oprindelige problem er for komplekst til at løse direkte. Masterproblemet skal i sidste ende give os den optimale løsning til vores oprindelige problem og er fra starten forholdsvis simpelt. Subproblemets opgave er nu løbende at tilføje "gode" delløsninger (søjler) til masterproblemet, indtil dets duale løsning er lovlig. I dette tilfælde siger almindelig dualitetsteori, at masterproblemet så er løst til optimalitet. Processen kan groft illustreres i Figur 2.



Figur 2: Skitse af processen bag søjlegenerering.

Figur 2 læses med start øverst. Her anvender vi Dantzig-Wolfe dekomposition (eller eventuelt andre relevante metoder) til at forberede vores oprindelige problemformulering, så den kan anvendes til søjlegenerering. Mere herom i afsnit 2.2. Dernæst opstilles og løses vores reducerede masterproblem. Hvis vores reducerede omkostninger er negative, er løsningen ikke optimal, da de duale variable er ulovlige, og vi sender derfor de duale variable ned til vores subproblem. I vores subproblem beregnes de modificerede objektionskoefficienter ud fra vores duale variable, og subproblemet løses. Dette giver os en løsning, som sendes tilbage til masterproblemet og bruges som en søjle i dette. Processen gentages, indtil masterproblemet er løst til optimalitet. ([6], s.188, 200)

2.2. Dantzig-Wolfe dekomposition

Udgangspunktet er et heltalsproblem P med et par særlige karakteristika. Lad A og D være koefficientmatricer med K søjler og lad $x \in \mathbb{Z}_+^n$ være vores variable. Vi ser da på følgende problem:

$$\begin{aligned} z &= \max \sum_{k=1}^K c^k x^k \\ \sum_{k=1}^K A^k x^k &= b \\ D^k x^k &\leq d_k, \quad k = 1, \dots, K \\ x^k &\in \mathbb{Z}_+^{n_k}, \quad k = 1, \dots, K \end{aligned}$$

Når vi udover dette har den egenskab, at løsningsrummet X for P kan opdeles i K begrænsede, uafhængige mængder

$$X^k = \{x^k \in \mathbb{Z}_+^{n_k} : D^k x^k \leq d_k\}, \quad k = 1, \dots, K,$$

har vi mulighed for at reformulere vores problem til brug sammen med søjlegenerering.

Målet med omformuleringen er at opnå en ny formulering P_R af problemet, der kan håndtere løsninger til P som søjler i P_R ; altså at hver løsning i P svarer til en søjle i P_R . Generelt ønsker vi at opstille denne omformulering på følgende måde:

$$\begin{aligned} z &= \max \sum_{k=1}^K \gamma^k \lambda^k \quad (P_R) \\ \sum_{k=1}^K B^k \lambda^k &= \beta \\ \lambda^k &\in \{0, 1\}^{n_k}, \quad k = 1, \dots, K, \end{aligned}$$

hvor B^k er en matrix, der har præcis én søjle for hver af de mange lovlige løsninger i X^k . λ^k er de tilhørende vektorer af beslutningsvariable, der afgør, om vi anvender de pågældende søjler i løsningen.

Vi ser nu igen på problemet P :

$$\begin{aligned} z &= \max \sum_{k=1}^K c^k x^k \\ \sum_{k=1}^K A^k x^k &= b \\ x^k &\in X^k \quad , \quad k = 1, \dots, K \end{aligned}$$

hvor $X^k = \{x^k \in \mathbb{Z}_+^n : D^k x^k \leq d_k\}$, $k = 1, \dots, K$. Vi indicerer nu de endeligt mange elementer i hver X^k med t : $\{x^{k,t}\}_{t=1}^{T_k}$, $x^{k,t} \in \mathbb{Z}_+^n$. Dette giver os mulighed for at omformulere X^k -mængderne på følgende måde (de er stadig identiske med de oprindelige mængder):

$$X^k = \left\{ x^k \in \mathbb{R}^{n_k} : x^k = \sum_{t=1}^{T_k} \lambda_{k,t} x^{k,t}, \sum_{t=1}^{T_k} \lambda_{k,t} = 1, \lambda_{k,t} \in \{0, 1\}, t = 1, \dots, T_k \right\}.$$

Her summerer vi over hver enkelt løsning i X^k og anvender λ -variablene til at udvælge netop ét af leddene. Da mængden indeholder alle mulige kombinationer heraf, vil vi få netop den oprindelige X^k -mængde ud af denne ikke-oplagte omskrivning.

Vi bruger nu omformuleringen af x^k til at omskrive P til et såkaldt *Integer Master Problem*:

$$\begin{aligned} z &= \max \sum_{k=1}^K \sum_{t=1}^{T_k} (c^k x^{k,t}) \lambda_{k,t} \quad (IPM) \\ \sum_{k=1}^K \sum_{t=1}^{T_k} (A^k x^{k,t}) \lambda_{k,t} &= b \end{aligned} \tag{1}$$

$$\sum_{t=1}^{T_k} \lambda_{k,t} = 1, \quad k = 1, \dots, K \tag{2}$$

$$\lambda_{k,t} \in \{0, 1\} \quad , \quad t = 1, \dots, T_k, \quad k = 1, \dots, K.$$

Bemærk, at vi har skiftet fra x^k til $\lambda_{k,t}$ som vores primære variable. Vi har desuden fået vores oprindelige problem omformuleret til den form, som vi ønskede. Denne omformulering kaldes ofte for en *Dantzig-Wolfe dekomposition*. I det følgende vil vi anvende denne formulering som basis for vores søjlegenerering. ([6], s.185ff)

2.3. Søjlegenereringsalgoritmen

Efter Dantzig-Wolfe dekompositionen er vi nu klar til at gå videre med trinene fra Figur 2. Vi opstiller derfor det *LP-relaxerede Masterproblem (LPM)* ved at relaxere vores λ -variable til $\lambda_{k,t} \geq 0$. I dette problem ser vi nu, at for hver af de oprindelige løsninger i

X^k har vi i LPM en søjle på formen

$$\begin{pmatrix} c^k x^{k,t} \\ A^k x^{k,t} \\ \mathbf{1}^k \end{pmatrix},$$

hvor $\mathbf{1}^k$ er en k -dimensionel 1-vektor. Vi kalder nu de duale variable hørende til (1) for $\{\pi_i\}_{i=1}^m$, og de duale variable hørende til (2) kalder vi $\{\mu_k\}_{k=1}^K$. Ideen er nu at løse vores masterproblem via god gammeldags simplex. Problemet herved er dog at finde den variabel, der skal skiftes ind, blandt de rigtig mange, der er at vælge imellem. Derfor bruger vi vores subproblem til at finde denne. Vi starter med at se på et reduceret masterproblem ($RLPM$) med et lille antal søjler, dog mindst én gyldig løsning for hvert subproblem. Vi bruger i det følgende $\sim$ til at betegne variable, koefficienter og værdier i dette reducere masterproblem.

$$\begin{aligned} \tilde{z} &= \max \tilde{c} \tilde{\lambda} & (RLPM) \\ \tilde{A} \tilde{\lambda} &= \tilde{b} \\ \tilde{\lambda} &\geq 0 \end{aligned}$$

Vi bemærker her, at (1) og (2) fra LPM er slået sammen i form af $\tilde{A}$ -matricen og $\tilde{b} = \begin{pmatrix} b \\ \mathbf{1} \end{pmatrix}$, og at $\tilde{c}$ og $\tilde{\lambda}$ er de tilhørende omkostninger og beslutningsvariable. Vi vil nu opstille de forskellige kriterier og trin, der kræves for at opbygge søjlegenereringsalgoritmen til løsning af IPM .

Feasibility af $RLPM$

Enhver løsning til $RLPM$ vil også være en løsning til LPM , da vi blot ser på et indsnævret løsningsrum med væsentlig færre søjler. Specielt vil den optimale løsning $\tilde{\lambda}^*$ til $RLPM$ også være en løsning til LPM , og vi kan derfor begrænse vores z ved hjælp af den duale løsning til $RLPM$:

$$\tilde{z} = \tilde{c} \tilde{\lambda}^* = \sum_{i=1}^m \pi_i b_i + \sum_{k=1}^K \mu_k \leq z. \quad (3)$$

Stopkriterium

LPM er løst til optimalitet, når alle vores reducerede omkostninger $c^k x^{k,t} - \pi A^k x^{k,t} - \mu_k$ er mindre end eller lig 0. Dette skal selvfølgelig undersøges for hver eneste af de mange søjler. I stedet for at gøre dette eksplicit, gør vi det implicit ved at opstille et subproblem:

$$\zeta_k = \max\{(c^k - \pi A^k)x - \mu_k : x \in X^k\}$$

Såfremt $\zeta_k = 0$ for alle $k = 1, \dots, K$, er vi kommet frem til, at vores duale løsning (π, μ) er lovlig for LPM . Fra almindelig dualitetsteori har vi dermed, at

$$z \leq \sum_{i=1}^m \pi_i b_i + \sum_{k=1}^K \mu_k.$$

Denne information koblet med (3) giver os så, at vores løsning $\tilde{\lambda}^*$ til LPM er optimal.

Generering af en ny søjle

Når vi står i den situation, hvor en af vores reducerede omkostninger ζ_k er skarpt større end 0, kan vi forbedre LPM . Dette gøres ved at introducere søjlen

$$\begin{pmatrix} c^k \tilde{x}^{k,t} \\ A^k \tilde{x}^{k,t} \\ \mathbf{1}^k \end{pmatrix}$$

i masterproblemet, svarende til løsningen $\tilde{x}^{k,t}$ i subproblemet ζ_k . Da vi anvender simplex til løsning af LPM , kan vi hurtigt re-optimere dette, da vi kun tilføjer en enkelt søjle.

En dual (øvre) grænse

Vi sætter nu $\zeta = (\zeta_1, \dots, \zeta_K)$. Hvis der gælder, at $(\pi, \mu + \zeta)$ er en lovlig dual løsning til LPM , så har vi en øvre grænse for LPM . $(\pi, \mu + \zeta)$ er en lovlig dual løsning, hvis $(c^k - \pi A^k)x - (\mu_k + \zeta_k) \leq 0$, for alle $x \in X^k$. Men da $\zeta_k = \max\{(c^k - \pi A^k)x - \mu_k : x \in X^k\}$, er dette oplagt opfyldt, og vi får dermed følgende øvre grænse på LPM :

$$z \leq \pi b + \sum_{k=1}^K \mu_k + \sum_{k=1}^K \zeta_k.$$

Alternativt stopkriterium

Vi kan komme i den situation, at løsningen $\tilde{x} = (\tilde{x}^1, \dots, \tilde{x}^K)$ til vores subproblem opfylder alle vores begrænsninger $\sum_{k=1}^K A^k \tilde{x} = b$ fra (1). I dette tilfælde vil løsningen også være optimal for LPM , da objektionsværdien i LPM har nået vores øvre grænse (udledt ovenfor):

$$z = \sum_{k=1}^K c^k \tilde{x}^k = \sum_{k=1}^K \pi A^k \tilde{x}^k + \sum_{k=1}^K \mu_k + \sum_{k=1}^K \zeta_k = \pi b + \sum_{k=1}^K \mu_k + \sum_{k=1}^K \zeta_k.$$

Ovenstående trin udgør byggestenene i søjlegenereringsalgoritmen til løsning af LPM . ([6], s.188f)

Hvis man ender med at få en optimal løsning til LPM , der ikke er binær, kan man anvende branch and bound til at opnå en heltallig løsning. Denne fremgangsmåde har vi valgt ikke at se nærmere på i dette projekt, men henviser til afsnit 11.4 i [6]. Vi skal i stedet anvende ovenstående algoritme i et eksempel.

2.4. Eksempel på søjlegenerering

I det følgende eksempel vil vi anvende søjlegenereringsalgoritmen til at løse en instans af det velkendte *symmetriske travelling salesman-problem* ($STPS$). Vi vil først anvende Dantzig-Wolfe dekomposition på den traditionelle formulering og derefter løse problemet ved hjælp af søjlegenerering.

Vi opskriver først problemet på dekomponérbar form. Lad der være givet en ikke-orienteret komplet graf $G = (V, E)$ med $|V| = n$ og $|E| = m$. Lad desuden hver kant $e \in E$ have vægt $c_e \in \mathbb{R}_+$, og lad $\delta(v)$ være de kanter, der går ud fra knude $v \in V$ i en

given løsning $x \in \mathbb{Z}_+^m$. *STSP* kan da formuleres på følgende måde:

$$\begin{aligned} z &= \min \sum_{e \in E} c_e x_e & (STSP) \\ \sum_{e \in \delta(v)} x_e &= 2, & \text{for alle } v \in V \\ x &\in X^1 \end{aligned} \tag{4}$$

Her har vi opdelt vores løsningsrum i netop én mængde X^1 :

$$X^1 = \left\{ x \in \mathbb{Z}_+^m : \sum_{e \in \delta(v_1)} x_e = 2, \sum_{e \in E(S)} x_e \leq |S| - 1, \emptyset \subset S \subseteq V \setminus \{v_1\}, \sum_{e \in E} x_e = n \right\}.$$

Denne mængde X^1 repræsenterer mængden af samtlige mulige 1-træer for problemet. Læg specielt mærke til, at vi i *STSP* kun har ét subproblem ($K = 1$), da enhver søjle vil repræsentere en komplet "løsning" til problemet. Vi fortsætter nu med selve Dantzig-Wolfe dekompositionen, så vi kan opnå en søjlebaseret formulering af problemet. Start med at omskrive x_e :

$$x_e = \sum_{t: e \in E^t} \lambda_t, \quad \text{for alle } G^t = (V, E^t), \quad t = 1, \dots, T_1.$$

Her repræsenterer G^t et af 1-træerne fra X^1 , med $E^t \subseteq E$. Lad desuden d_v^t være knudevalensen af knude v i 1-træet G^t . Med denne omskrivning af x_e bliver vores begrænsninger (4) nu til:

$$\sum_{e \in \delta(v)} x_e = \sum_{e \in \delta(v)} \sum_{t: e \in E^t} \lambda_t = \sum_t d_v^t \lambda_t = 2, \quad \text{for alle } v \in V.$$

Efter denne dekomposition kan vi nu opskrive vores LP-relaxerede Master Problem for *STSP*:

$$z = \min \sum_{t=1}^{T_1} (cx^t) \lambda_t \quad (LPM - STSP)$$

$$\sum_{t=1}^{T_1} d_v^t \lambda_t = 2, \quad \text{for alle } v \in V \tag{5}$$

$$\sum_{t=1}^{T_1} \lambda_t = 1 \tag{6}$$

$$\lambda \in \mathbb{R}_+^{T_1}$$

Vi bemærker nu, at i ethvert 1-træ vil knudevalensen af knude v_1 være lig 2, dvs. $d_{v_1}^t = 2$ for $t = 1, \dots, T_1$. Dette betyder, at (5) blandt andet indeholder følgende begrænsninger:

$$\sum_{t=1}^{T_1} 2\lambda_t = 2, \text{ hvilket vil sige } \sum_{t=1}^{T_1} \lambda_t = 1, \quad t = 1, \dots, T_1.$$

Men disse begrænsninger er præcis magen til (6), som derfor er redundante.

Til *LPM-STSP* tilknytter vi nu duale variable. Til (5) knytter vi $\{\pi_v\}_{v \in V}$, og da (6) er redundant, undlader vi at tilknytte den duale variabel μ hertil. Hvis vi ser på reducerede omkostninger for 1-træet G^t og lader x_e^t beslutte, om vi bruger kanten $e \in E$ i 1-træet t , får vi vores reducerede omkostninger for *LPM-STSP* til:

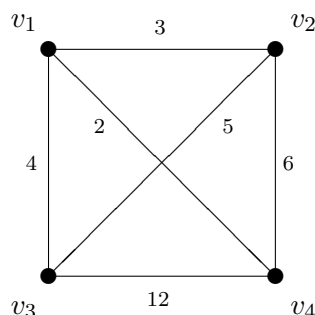
$$cx^t - \sum_{v \in V} d_v^t \pi_v = cx^t - \sum_{v \in V} \pi_v \sum_{e \in \delta(v)} x_e^t = \sum_{e \in E} (c_e - \pi_v - \pi_w) x_e^t.$$

Her er $e = (v, w)$, altså kanten $e \in E$ går fra knude v til knude w . Ud fra disse reducerede omkostninger kan vi nu opskrive vores (ene) subproblem:

$$\zeta_1 = \min \left\{ \sum_{e \in E} (c_e - \pi_v - \pi_w) x_e : x \in X^1 \right\}. \quad ([6], \text{ s.190f})$$

Vi ser nu på en instans² af *STSP* med følgende afstandsmatrix:

$$c_e = \begin{pmatrix} - & 3 & 4 & 2 \\ - & - & 5 & 6 \\ - & - & - & 12 \\ - & - & - & - \end{pmatrix}$$



Vi starter med at udvælge en initial løsning $x_{12} = x_{14} = x_{23} = x_{34} = 1$ samt to tilfældige 1-træer til vores masterproblem:

$$\begin{aligned} z = \min \quad & 22\lambda_1 + 23\lambda_2 + 24\lambda_3 \\ & 2\lambda_1 + 2\lambda_2 + 2\lambda_3 = 2 \\ & 2\lambda_1 + 1\lambda_2 + 2\lambda_3 = 2 \\ & 2\lambda_1 + 3\lambda_2 + 3\lambda_3 = 2 \\ & 2\lambda_1 + 2\lambda_2 + 1\lambda_3 = 2 \\ & \lambda \in \mathbb{R}_+^4 \end{aligned}$$

Vi løser nu dette problem i GAMS (se kildekode i Appendix B, side 46). Løsningen finder vi til at være $\lambda = (1, 0, 0)$ med $z = 22$ og duale variable $\pi = (11, 0, 0, 0)$. Vi opskriver nu vores matrix med reducerede omkostninger for at kontrollere, om vores løsning er optimal:

$$r_1 = \begin{pmatrix} - & -8 & -7 & -9 \\ - & - & 5 & 6 \\ - & - & - & 12 \\ - & - & - & - \end{pmatrix}$$

²Exercise 11.8.1 fra [6].

Vi opskriver og finder nu løsningen til vores subproblem ζ_1 og får:

$$\zeta_1 = \min\{2, -4, 3, 0, -6, 1, 1, -5, 2\} = -6$$

Vores løsning er dermed ikke optimal, og vi tilføjer derfor 1-træet svarende til løsningen til subproblemet $x_{12} = x_{14} = x_{23} = x_{24} = 1$ med længde 16 som søjle i vores masterproblem. Dette giver os nu en ny nedre grænse:

$$22 + \zeta_1 = 22 - 6 = 16 \leq z \leq 22.$$

Vores nye masterproblem bliver nu:

$$\begin{array}{llllll} z = \min & 22\lambda_1 & +23\lambda_2 & +24\lambda_3 & +16\lambda_4 & \\ & 2\lambda_1 & +2\lambda_2 & +2\lambda_3 & +2\lambda_4 & = 2 \\ & 2\lambda_1 & +1\lambda_2 & +2\lambda_3 & +3\lambda_4 & = 2 \\ & 2\lambda_1 & +3\lambda_2 & +3\lambda_3 & +1\lambda_4 & = 2 \\ & 2\lambda_1 & +2\lambda_2 & +1\lambda_3 & +2\lambda_4 & = 2 \\ \lambda \in & \mathbb{R}_+^4 & & & & \end{array}$$

Løsningen til dette nye masterproblem bliver $\lambda = (0, \frac{1}{2}, 0, \frac{1}{2})$ med $z = \frac{39}{2}$. Vi beregner igen de reducerede omkostninger ud fra de duale variable $\pi = (\frac{25}{4}, 0, \frac{7}{2}, 0)$:

$$r_1 = \begin{pmatrix} - & -\frac{13}{4} & -\frac{23}{4} & -\frac{17}{4} \\ - & - & \frac{3}{2} & 6 \\ - & - & - & \frac{17}{2} \\ - & - & - & - \end{pmatrix}$$

og finder $\zeta_1 = -\frac{5}{2}$, og at vores løsning dermed ikke er optimal. Igen tilføjer vi en ny søjle svarende til subproblemet's løsning: $x_{13} = x_{14} = x_{23} = x_{24} = 1$, samt finder en ny nedre grænse

$$\frac{39}{2} + \zeta_1 = 17 \leq z \leq 22$$

Eftersom dette 1-træ også er en tur, følger det af det alternative stopkriterium, at løsningen er optimal. Alternativt kan man også se, at vores nye søjle har længde 17.

3. Problem og modeller

I det følgende afsnit vil vi beskrive S-togsnettet generelt og de forskellige egenskaber og de krav, der stilles til det. Herunder blandt andet hvad der kan gøres, hvis kravene ser ud til ikke at kunne opfyldes. Desuden vil vi se på de forskellige typer af større og mindre forstyrrelser, der typisk kan forekomme på et banenet som det, S-togene kører på dagligt. Dernæst vil vi introducere det primære problem for dette projekt: genindsættelse af aflyste toglinier, som vi herefter vil se på to modeller for. Afslutningsvist vil vi sammenligne disse to modeller.

3.1. S-togsnettet generelt

DSB A/S er et af de 16 selskaber, der driver jernbanedrift på det danske skinnenet. DSB beskæftiger sig med både lokal, national og international passagertransport. DSB S-tog er et underselskab af DSB, som driver S-togsnettet i København og omegn. S-togsnettet består af 7 linier: A, B, Bx, C, E, F og H. Linierne passerer mellem 12 og 31 stationer hver, og togene kører generelt i 20-minuttersdrift ved hver station. I myldretiden vil de fleste linier køre i 10-minuttersdrift; desuden er en enkelt af linierne (Bx) kun i brug i myldretiden. Her skelnes der mellem linierne ved at kalde myldretidstogene (10-minuttersdrift) for plus-linier. I myldretiden kører der derfor både en A og en A+ linie. S-togsnettets linier kan ses i Figur 3 herunder.



Figur 3: Liniekort for S-togsnettet i København (2009)

Som det kan ses af liniekortet, passerer langt de fleste af linierne det centrale København, specielt mellem stationerne Dybbølsbro og Svanemøllen. Hver S-togslinie har tilknyttet

4-10 tog. Antallet afhænger af, hvor lang tid det tager et tog at køre fra den ene endestation til den anden og tilbage igen (herefter forkortet RTT_l). Linie l skal altså som udgangspunkt bruge $\left\lceil \frac{RTT_l}{20} \right\rceil$ tog for at opretholde en 20-minuttersdrift. To linier kan dog godt ”dele” et tog, så det for eksempel i myldretiden skiftevis kører almindelig og +.

Der køres med to typer af togsæt: Litra SE med en længde på 42,58 meter, og Litra SA med en længde på 83,78 meter. Disse to togtyper kan sammenkobles. På grund af længden af de forskellige perroner på stationerne kan der dog maksimalt være enten to SA-togsæt eller to SE-togsæt plus ét SA-togsæt koblet sammen.

Hvert tog på nettet tildeles et tognummer. Dette nummer er 5 cifre langt og indeholder forskellige informationer. Hvert tognummer er på formen L-D-R-KH, hvor L angiver linienummeret (1-7), D angiver, om der er tale om et almindeligt eller et plus-tog (1 eller 5), R angiver retningen i forhold til passage af København H (nordgående eller sydgående), og KH angiver det tidsinterval på dagen, hvor toget ankommer til København H. Dette tidsinterval aflæses som KH modulo 3. Da der køres i 20-minuttersdrift, er der 3 tidsintervaller per time. Ved at regne modulo 3 får vi timen på dagen, resten giver os hvilket af de 3 mulige tidsintervaller, der er tale om. Vi illustrerer med et eksempel: tog nummer 41225 kører på linie C (4) som et almindeligt tog (1) i sydgående retning (2) mod Ballerup og ankommer til København H som det andet tog efter klokken 8 ($25/3 = 8$, med rest 1)

Hver gang et tog når en endestation, vil toget vende om, og tognummeret vil ændre sig. Et tog, der kører samme linie en hel dag, vil altså være forsynet med mange tognumre. En sådan sekvens af tognumre for en hel dag kaldes en togstreng, og er altså en lang streng af tognumre. I Tabel 1 ses et eksempel på en togstreng for linie C. ([4], s.2ff).

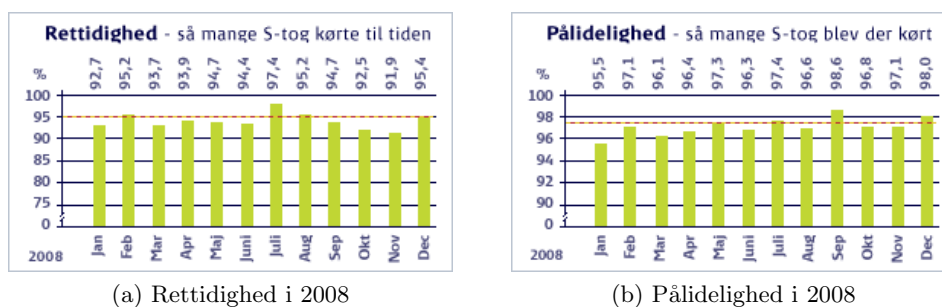
41116	41219	41122	41225	41128	41231	...	41171	41202
-------	-------	-------	-------	-------	-------	-----	-------	-------

Tabel 1: Eksempel på togstreng for linie C.

3.2. Krav og forstyrrelser

Al jernbanedrift i Danmark er underlagt Transportministeriets regler, og DSB S-tog er i deres kontrakt forpligtet til at overholde visse driftsmæssige krav og mål for at undgå økonomiske sanktioner eller bøder. Overordnet er der to krav, der skal opfyldes: rettidighed og pålidelighed. Rettidigheden stiller krav om, hvor mange tog der skal køre til tiden. ”Til tiden” vil i S-togssammenhæng sige, at et tog er mindre end 2 minutter og 29 sekunder forsinket. Rettidighed måles i procent af de tog, der kommer til tiden. Pålidelighed måles i procent af de planlagte afgang, der faktisk kommer afsted. Undtagelser hertil kunne eksempelvis være, når en hel toglinie udtages pga. en forstyrrelse (som er emnet for analysen i afsnit 3.3) eller et tog, der går i stykker. Den stiplede linie, som kan ses i Figur 4 svarer til S-togs målsætning, som er identisk med kravene i kontrakten med Transportministeriet: 95% for rettidighed og 97,5% for pålidelighed.

Man kan nu undre sig over, hvorfor det mon ikke er muligt at opnå 100% rettidighed og pålidelighed. Grunden til dette er selvfølgelig, at der kan opstå forsinkelser eller aflysninger af et væld af årsager. Disse kan være udefra kommende eller være forårsaget af dårlig eller for stram planlægning. Udefra kommende forsinkelser kan f.eks. skyldes uheld



Figur 4: Data for rettidighed og pålidelighed for S-togsdriften i 2008.

(nedfaldne køreledninger, væltede træer, etc.), signalproblemer, passagerforsinkelser (for eksempel hjælp til ombordstigning for handicappede), eller planlagt vedligeholdelse (som BaneDanmark tager sig af).

Forsinkelser kan ligeledes relateres til internt opståede situationer eller fejl, såsom tekniske fejl på togene, forsinkede medarbejdere, generelle planlægningsfejl, manglende togsæt- eller personalekapacitet pga. for stram planlægning eller lignende.

Der findes en del forskellige måder at løse forsinkelsesproblemer på, alt efter graden af forsinkelsen eller forstyrrelsen.

Ekstra tid i køreplanen

I køreplanerne er der indlagt en vis mængde ekstra tid langs stationerne på nettet. På en strækning, der kan køres på 35 minutter, kan der f.eks. være afsat 38 minutter, ligesom der i endestationerne typisk er ekstra tid mellem ankomst og afgang af samme togsæt. På den centrale skinnenet omkring København H kommer togene med ca. to minutters interval, så her er der ikke de store muligheder for at indhente forsinkelser på denne måde.

Ruteændringer

Hvis et tog, der ikke stopper på alle stationer (f.eks. på linie Bx) indhenter et tog, der stopper på alle stationer (f.eks. på linie B), kan det forreste tog konverteres til et Bx-tog og det bagerste til et B-tog. Dette gøres i praksis på førstkommende station, hvor begge tog stopper, men vil kræve at eventuelle passager kan nå at skifte tog. Desuden kan et tog på en hvilken som helst linie tvinges til at vende, før det har nået endestationen. Igen skal der gives mulighed for passagerflytninger til andre tog. Sidst, men ikke mindst, kan enkelte stationer overspringes undervejs på en linie.

Overhalinger

Et "langsomt" tog, som stopper på alle stationer, kan trække ind på et sidespor og give plads for et "hurtigt" tog, der ikke stopper på alle stationer.

Indsættelse af ekstra tog

På endestationerne kan eventuel ekstra kapacitet indsættes som erstatning for et forsinket tog, der ikke kan nå frem og vende i tide. Det forsinkede tog udtages herefter. Dette er en neutral løsning fra passagerens synspunkt, men er omkostningsmæssigt ret dyr, da der skal ekstra personale ud til det ekstra tog.

Aflysning af en hel linie

For at skabe ekstra plads på skinnenettet kan det blive nødvendigt at udtage en hel toglinie i en vis periode. Ved at trække disse tog ud, er der mere plads til, at de resterende tog på nettet kan indhente den opståede forsinkelse. Dette vil som oftest ske i myldretiden, hvor der er mindst plads på skinnenettet, og den udtagne linie vil i langt de fleste tilfælde være en plus-linie.

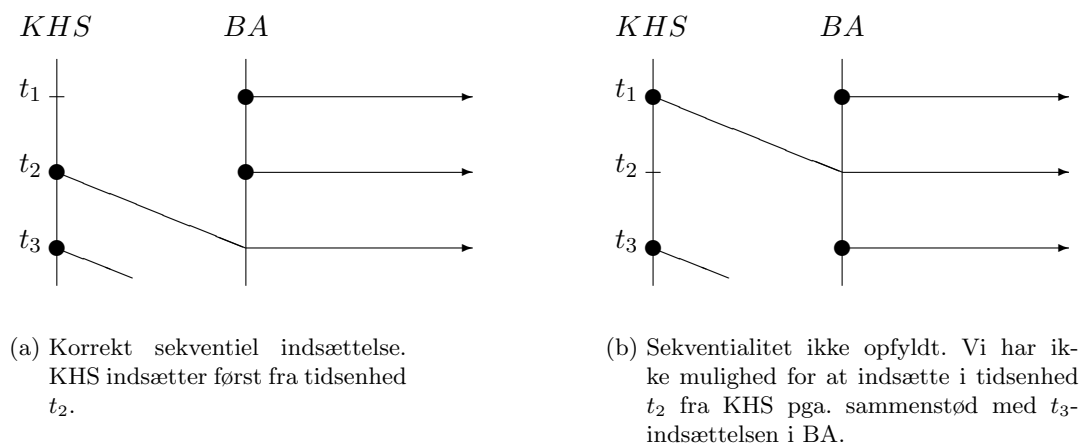
På nær den sidste er ovenstående muligheder forholdvis små ændringer i køreplanen. Ved aflysning af en hel linie opstår der store gener og udfordringer for både passagerer, planlæggere og ansatte. For passagererne betyder det, at der kun kører halvt så mange tog på den givne linie. Lokomotivførerne skal transporteres til personaledepotet på København H, og planlæggerne skal sørge for at få hele logistikken omkring ændringerne til at gå op. Desuden er det økonomisk dyrt for S-tog at have medarbejdere, der ikke kan lave noget og tog, der står stille. Jo længere aflysningen varer ved, jo større omkostninger. ([4], s16ff, [2], s13ff)

3.3. To mulige modeller

Vi vil nu præsentere og gennemgå de to modeller til løsning af genindsættelsesproblemet, som S-tog selv har udviklet. Oprindeligt blev model 1 udviklet med henblik på løsning af problemet. Under implementationen opstod ideerne til model 2, som derfor ligeledes blev udviklet. Vi vil dog først se på lidt flere af de forudsætninger, som problemet løses under. Når en toglinie udtages, foregår det ved, at de implicerede togsæt rangeres ud på depoter undervejs på strækningen. Desuden udtages et antal tog på endestationerne. Normalt vil et togsæt blive udtaget på førstkommende depot på linien. Når BaneDanmark giver lov til at genindsætte linien (når problemet, der skabte udtagelsen, er blevet løst), får S-tog besked på at genoptage driften af linien. Opgaven går nu ud på hurtigst muligt at få genindsat den udtagne linie med mindst mulig gene for passagerer og ansatte. I den forbindelse er der fire faktorer, der skal tages højde for.

1. For at kunne indsætte et tog fra et depot, skal der selvfølgelig være en lokomotivfører til stede. Hos S-tog er der kun ét personaledepot, som ligger på København H. Her sendes alle lokomotivførerne fra de udtagne tog hen. Dermed skal de enkelte lokomotivførere transporteres ud til depoterne igen fra København H, før togene kan genindsættes. Når dette gøres, skal der desuden tages højde for personalemæssige regler og overenskomster, såsom pauser, vagtplaner og lignende. Dette ligger dog ikke i vores problem, men efterfølgende hos personaleplanlæggerne.
2. Alle de tog, der er blevet udtaget til et bestemt depot, skal også genindsættes fra dette depot igen, og der kan selvfølgelig ikke genindsættes flere tog, end der er tilgængelige på depotet. Det samlede antal skal selvfølgelig harmonere, da det er en hel linie, der tages ud og sættes ind igen.
3. Der kan ikke indsættes et tog et sted i køreplanen, hvor der ikke er et "hul". Her skal der både tages højde for de tog, der allerede kører, samt for dem, der skal indsættes. Med andre ord: det skal sikres, at hver togstreng på linien tildeles netop ét tog. Hvis der f.eks. indsættes et tog på det foregående depot D_1 , og der er 20 minutter imellem depoterne D_1 og D_2 , kan vi ikke indsætte et nyt tog i den efterfølgende tidsenhed fra depot D_2 .

4. Så snart vi begynder at indsætte tog fra et depot igen, skal der for passagerernes vedkommende være tog efter køreplanen (10- eller 20-minutters) uden undtagelse fra første indsættelse af et udtaget tog. I Figur 5 er givet et eksempel (fra linie C), hvor denne *sekventialitet* ikke er opfyldt.



Figur 5: Eksempler på sekventialitet. Der skal indsættes 2 tog fra KHS og 2 fra BA.

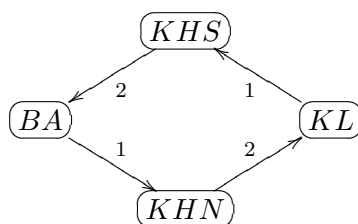
De to modeller repræsenterer hver deres måde at løse genindsættelsesproblemet på. Den første er et minimeringsproblem, der tilstræber at minimere de sidste to cifre (KH) i det sidst indsatte togs tognummer. Modellen bygger på løsning af et standard MIP-problem og er den mest komplekse med hensyn til variabelantal og begrænsninger. Den anden forsøger at minimere togindsættelsestiden ved at tildele en højere straf i objekt-funktionen for tog, jo senere de indsættes. Modellen kræver et større forarbejde af ind-data, men er til gengæld forholdsvis simpel i forhold til antal variable og begrænsninger.

Vi starter med at definere to mængder af depoter, D og K . D er mængden af alle *depoter* på den i problemet givne toglinie; her er der ikke angivet køreretning, men blot fysiske depoter. Eksempelvis for linie C er der tale om $D = \{BA, KH, KL\}$ (Ballerup, København H, Klampenborg³). Lad desuden K være mængden af alle *retningsdepoter*, dvs. alle depoter inklusive dobbeltrettede depoter (dem som ikke er endestationer). $K = K^E \cup K^I$, hvor K^E er de to endestationer (BA og KL for linie C), og K^I er resten med retning (dvs. de gennemkørende depoter KHN og KHS). Lad desuden d_k være antallet af tog, der skal indsættes fra depot k . Her kan k komme fra enten D eller K , alt efter konteksten. $t \in T$, $T = \{1, 2, \dots, t_{max}\}$ angiver den mængde af tidsenheder, som vi kan vælge at sætte tog ind i. t_{max} vælges passende stor kan med fordel vælges som summen af den tid, det tager for lokomotivførere at komme ud til fjerneste depot, plus det samlede antal tog, der skal genindsættes på linien.⁴ ([3], s.14)

³I retning fra sydligste over København H til nordligste. For linien C^+ indgår Frederikssund også på linien, altså for C^+ er $D = \{FS, BA, KH, KL\}$

⁴I [3] indføres ligeledes mængden $L = \{N, S\}$, som angiver de to retninger, der kan indsættes tog i fra depoter, som ikke er endestationer. Da vi for S-togs vedkommende altid kun har to retninger, har vi valgt at implementere dette "hårdt" i modellerne.

I Figur 6 er vist linie C og de 6 tog, der skal til for at dække linien. Tallene angiver, hvor mange tidsenheder af 20 minutter, der går, fra toget afgår fra forrige depot, og indtil det afgår fra det næste.



Figur 6: Linie C og afstanden i tidsenheder mellem depoterne

3.4. Model 1

Vi skal nu se på den første af de to modeller, som er mest kompleks med hensyn til variable og begrænsninger, men hvor input er forholdsvis simpelt. Modellen gør brug af beslutningsvariablene $x_{i,t,k} \in \{0,1\}$, hvor $i \in I$ angiver hvilken togstreng i , som vi skal indsætte et tog i ; $t \in T$ angiver hvilket tidsenhed toget indsættes i ; $k \in K$ angiver, hvilket depot toget skal indsættes fra. $x_{i,t,k}$ er altså lig 1, hvis et tog indsættes i togstreng i , i tidsenhed t fra depot k , og er 0 ellers.

I denne model er der et forholdsvis stort antal begrænsninger, der skal være overholdt. Vi venter med objektfunktionen til sidst, da beskrivelsen af den afhænger af en god forståelse af de forskellige begrænsninger.

$$\sum_{t,k} x_{i,t,k} = 1, \quad \forall i \in I \quad (7)$$

$$\sum_i x_{i,t,k} \leq 1, \quad \forall t \in T, \forall k \in K \quad (8)$$

(7) sikrer, at hver togstreng tildeles netop en gang. Begrænsning (8) sikrer, at der maksimalt indsættes ét tog fra hvert depot i hver tidsenhed.

Vi kommer nu til de begrænsninger, der sikrer, at antallet af tog, der bliver sendt af sted fra hvert depot, svarer til det antal, der rent faktisk blev taget ud. Denne model sikrer også, at der bliver sendt lige mange tog afsted i hver retning fra hvert gennemkørende depot. Der er overordnet set to grunde til dette. For det første har man altid gjort sådan og har haft nogenlunde succes med det. For det andet er der et hensyn til passagererne gemt i det: hvis en passager på en station ser, at togene nu kører normalt i den ene retning men ikke i den anden, giver det frustration. Derfor introducerer vi nu variablene d_k^I , $k \in K^I$ til at være antallet af tog indsat i hver retning på hvert gennemkørende

depot. Vi udmønter disse krav i følgende begrænsninger:

$$\sum_{i,t} x_{i,t,k} = d_k, \quad \forall k \in K^E. \quad (9)$$

$$\sum_{i,t,k} x_{i,t,k} = \sum_k d_k, \quad k \in K^I \quad (10)$$

$$\sum_{i,t} x_{i,t,k} = d_k^I, \quad \forall k \in K^I \quad (11)$$

$$d_k^I \geq \left\lfloor \frac{d_k}{2} \right\rfloor, \quad \forall k \in K^I \quad (12)$$

$$d_k^I \leq \left\lceil \frac{d_k}{2} \right\rceil, \quad \forall k \in K^I \quad (13)$$

(9) sikrer, at antallet af tog, der indsættes fra *endedepoterne*, svarer til det antal, der blev taget ud der. (10) sikrer at antallet af indsatte tog på de *gennemkørende* depoter svarer til det antal, der er blevet udtaget fra dem. Grunden, til at vi ikke kan se på de enkelte retningsdepoter, er, at vi endnu ikke kender antallet, der skal sættes ind i hver retning. Begrænsningerne (11), (12) og (13) hører sammen og sikrer, at der bliver sendt lige mange tog i hver retning. Hvis der er tale om et ulige antal tog, der skal indsættes, for eksempel 5, sendes der 2 i den ene retning og 3 i den anden retning. Her sikrer (11) i vores eksempel, at der ikke sendes 3 tog i hver retning (hvilket er umuligt).

Vi introducerer nu endnu en konstant, $in_{i,t,k}$, der beskriver, hvornår hver togstreng er på de forskellige depoter. Med andre ord: hvis et tog fysisk kan indsættes i togstreng i fra depot k i tidsenheden t , er $in_{i,t,k} = 1$, og 0 ellers. Dette giver anledning til følgende begrænsning, der sikrer, at vi ikke sætter et tog ind, som vi ikke har mulighed for at sætte ind:

$$x_{i,t,k} \leq in_{i,t,k}, \quad \forall i \in I, j \in J, k \in K \quad (14)$$

Vi indfører heltalsvariablene $start_k$ og $slut_k$, $k \in K$. $start_k$ definerer første tidsenhed, hvor der indsættes tog fra retningsdepot k ; $slut_k$ angiver så sidste tidsenhed, hvor der indsættes tog fra retningsdepot k . Følgende begrænsning sikrer, at afstanden mellem $start_k$ og $slut_k$ er lig det antal tog, der skal indsættes fra retningsdepot k . Dermed vil togene fra de enkelte retningsdepoter blive indsat sekventielt:

$$start_k + \sum_{i,t} x_{i,t,k} - 1 = slut_k, \quad \forall k \in K. \quad (15)$$

Vi kigger nu på krav nummer 1 fra afsnit 3.3, altså at vi på hvert retningsdepot k først begynder at indsætte tog, når vi har fået lokomotivførere derud. Derfor indfører vi konstanterne $clok_k$, $k \in K$. Disse siger for hvert retningsdepot, hvor mange tidsenheder vi skal vente, før en lokomotivfører kan ankomme. $start_k$ skal derfor sættes større end dette:

$$start_k \geq clok_k + 1, \quad \forall k \in K. \quad (16)$$

Til sidst skal vi have fastsat selve værdierne for $start_k$ og $slut_k$. Konstanten M vælges

tilpas stor:

$$start_k \leq t + M \cdot (1 - x_{i,t,k}), \quad \forall i \in I, t \in T, k \in K \quad (17)$$

$$slut_k \geq t - M \cdot (1 - x_{i,t,k}), \quad \forall i \in I, t \in T, k \in K \quad (18)$$

Det, vi ønsker at minimere i objektværdien, er det højeste tognummer, og vi introducerer derfor to yderligere variable $tognummer_{i,t,k}$ og $maxtognummer$, samt de to konstanter $initialtog$ og $ctog_k$, $k \in K$. $tognummer_{i,t,k}$ er det femcifrede tognummer (jfr. afsnit 3.1), som blandt andet beskriver det tidsinterval, hvor toget passerer KBH, samt retningen på toget. $maxtognummer$ er et tocifret tal, der fastsættes ud fra det senest indsatte togs nummer (baseret på de sidste to cifre i tognummeret). $initialtog$ er tognummeret på det første tog, der skal transportere lokomotivførere ud til et eller flere af depoterne. $ctog_k$ er en teknisk konstant, der tilføjes for at korrigere tognumrene på de enkelte depoter, så der indsættes med korrekte tognumre alt efter indsættelsestidspunkt og retning. $tognummer_{i,t,k}$ og $maxtognummer$ beregnes derfor på følgende måde:

$$tognummer_{i,t,k} = (initialtog + ctog_k + t) \cdot x_{i,t,k}, \quad \forall i \in I, t \in T, k \in K \quad (19)$$

$$maxtognummer \geq tognummer_{i,t,k} \pmod{100}, \quad \forall i \in I, t \in T, k \in K \quad (20)$$

Vi er nu endelig nået frem til modellens objektfunktion. Denne er ganske simpel og søger simpelthen at minimere det største tognummer (hvorved vi får genindsættelsen overstået tidligst muligt):

$$\min \quad maxtognummer \quad (21)$$

([3], s12ff)

3.5. Model 2

Vi vil nu rette vores fokus på den anden af de to modeller, der bygger på koncepter fra søjlegenerering. Modellen er karakteriseret ved et forholdsvis komplekst input men få og ukomplicerede begrænsninger. Input til modellen består af to matricer, A og B , samt en række objektfunktionskoefficienter c_j , hvor $j \in J$ er en af alle de mulige indsættelser af ét tog, vi har at gøre med.

$A = [a_{k,t,j}]$, $a_{k,t,j} \in \{0, 1\}$ er en tredimensional matrix: $k \in K$ angiver retningsdepotet; $j \in J$ definerer en af de mulige indsættelser af tog.

Eksempelvis vil et 1-tal i $a_{KHS,3,21}$ betyde, at der afgår et tog på København H i sydgående retning i tidsenhed 3, hvilket svarer til indsættelse nummer 21.

	BA						KHN						KL						KHS					
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
BA	t_1																							
	t_2																							
	t_3	1																	1					
	t_4		1																	1				
	t_5			1																	1			
	t_6				1								1									1		
	t_7					1								1									1	
	t_8						1								1									1
KHN	t_1						1																	
	t_2							1																
	t_3								1															
	t_4	1								1									1					
	t_5		1								1									1				
	t_6			1								1									1			
	t_7				1								1									1		
	t_8					1								1									1	
KL	t_1							1						1										
	t_2								1						1									
	t_3									1						1								
	t_4										1						1							
	t_5											1						1						
	t_6	1											1						1					
	t_7		1																	1				
	t_8			1																	1			
KHS	t_1																		1					
	t_2																			1				
	t_3																				1			
	t_4							1						1								1		
	t_5								1						1								1	
	t_6									1						1								1
	t_7	1									1						1							
	t_8		1									1						1						

I ovenstående skema er A -matricen tegnet for linie C, hvor der skal genindsættes 6 tog. Læg mærke til, at første indsættelse fra depot BA kan ske i tidsenhed t_3 , da der kræves mindst 2 tidsenheder for at få en lokomotivfører fra mandskabsdepotet og ud til Ballerup station. Vælger vi en indsættelse på BA i tidsenhed t_3 ($x_1 = 1$), vil dette tog derfor være på KHN i tidsenhed t_4 , KL i tidsenhed t_6 og i KHS i tidsenhed t_7 . Blokkene i matricen er inddelt forskelligt, alt efter hvilken linie der er tale om. Vi kan teoretisk set komme ud i det tilfælde, at vi er nødt til at indsætte alle 6 tog fra ét depot.

Den første *bloksøjle* tager udgangspunkt i mulige indsættelser i Ballerup, den anden i København Nord osv. Et 1-tal i $a_{KHS,7,1}$ vil derfor betyde, at et tog er indsat i Ballerup, og er i København H i sydgående retning i tidsenhed 7, hvilket svarer til indsættelse nummer 1.

B -matricen angiver, hvilket depot $k \in D$ (dvs. uden retning) vi indsætter tog fra i de enkelte søjler og er forholdsvis simpel:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
BA	1	1	1	1	1	1																		
KH							1	1	1	1	1	1								1	1	1	1	1
KL													1	1	1	1	1	1						

Modellens beslutningsvariable $x_j \in \{0,1\}$, $j \in J$, angiver, om vi vælger at anvende indsættelsen i søjle j eller ej, og som simpel objektfunktion anvendes følgende:

$$\min z = \sum_j c_j x_j, \quad j \in J \quad (22)$$

Objektfunktionskoefficienterne c_j tildeler en forholdsmæssig straf, der er afhængig af, hvor sent toget indsættes fra det pågældende depot. For eksempel vil $c_1 = 3, c_2 = 4, \dots, c_6 = 8, c_7 = 1, \dots$ Vi ønsker dermed selvfølgelig at undgå sene indsættelser.

Udover objektfunktionen er der tre begrænsninger, der skal overholdes:

$$\sum_j a_{k,t,j} \cdot x_j \leq \sum_j a_{k,t+1,j} \cdot x_j, \quad \forall k \in K, \forall t \in \{1, 2, \dots, t_{max} - 1\} \quad (23)$$

$$\sum_j a_{k,t,j} \cdot x_j \leq 1, \quad \forall k \in K, \forall t \in T \quad (24)$$

$$\sum_j b_{k,j} \cdot x_j = d_k, \quad \forall k \in D \quad (25)$$

$$x_j, b_{k,j}, a_{k,t,j} \in \{0, 1\}.$$

(23) sikrer, at togene indsættes sekventielt i køreplanen, som beskrevet i punkt 4 i afsnit 3.3. Det sikres altså, at der altid kommer et tog bagefter det indsatte. (24) sikrer, at der ikke kører to tog fra samme station i samme retning i samme tidsenhed. (25) sikrer til sidst, at antallet af indsatte tog per depot svarer til det antal, der er blevet taget ud på det pågældende depot.

3.6. Sammenligning af modeller

Vi skal nu forsøge at sammenligne de to modeller gennemgået ovenfor. Vi skal se på forskellige aspekter i de to modeller: deres løsningsmetode (MIP og søjlegenerering); måden hvorpå de håndhæver de krav, der er angivet i afsnit 3.3; hvilket output de to modeller giver til videre brug, samt deres kompleksitet mht. at finde en brugbar (eller optimal) løsning.

Først ser vi på den generelle struktur af modellerne, eller hvordan de i operations-analytisk forstand løses. Model 1 er et nogenlunde ”standard” -MIP: der opstilles 14 sæt begrænsninger og en objektfunktion, som anvendes på forholdsvis simple inddata. Blandt andet bruges 3 sæt begrænsninger til at sikre, at der sendes ”cirka” lige mange tog i hver retning fra hvert depot, og 4 sæt til at fastsætte $start_k$ - og $slut_k$ -variablene. Overordnet kan man sige, at i denne model ligger ”arbejdet” i at opstille ens begrænsninger så generelt, at modellen er nem at udbrede til flere toglinier, eller udvidelser med flere krav (mere herom i afsnit 4). De eneste inddata, der ikke trivielt kan beregnes, er $in_{i,t,k}$ -matricen samt konstanterne $initialtog$ og $ctog_k$, $k = 1, \dots, K$, som afhænger af genindsættelsestidspunktet.

Model 2 bygger på og kan med fordel løses ved hjælp af søjlegenerering, som beskrevet i afsnit 2. Hvor der i model 1 afgøres med beslutningsvariablene $x_{i,t,k}$, om vi indsætter et fysisk tog i togstreng i i tidsinterval t fra depot k , er der i model 2 tale om valg af søjler (x_j), der hver svarer til bestemte indsættelser. Da antallet af søjler i S-togs tilfælde er relativt lavt, kan problemet løses på normal vis uden brug af søjlegenerering. Man kan dog sagtens forestille sig større scenarier med afskillige hundrede søjler, hvor teorien fra afsnit 2 vil skulle anvendes. For eksempel hvis man udviklede en lignende model, der indeholdt *hele* S-togsnettet. I denne model er der desuden et væsentlig lavere antal begrænsninger. Vi kan her nøjes med 3 sæt begrænsninger og selvfølgelig en objektfunktion. Der kræves dog til gengæld stort forarbejde i at forberede inddata til modellen.

I model 2 er inddata arrangeret således, at togenes fysiske placeringer og afgang følger rundt på linien. Hvis man f.eks. vælger en indsættelse i Ballerup, vil der i samme

søjle stå, i hvilket tidsinterval dette tog afgår fra de andre stationer på linien (vel at mærke kun inden for grænsen af tidsintervaller, t_{max}). Det er denne ekstra information, der komplicerer udarbejdelsen af inddata til modellen. Dette er ikke tilfældet i model 1, hvor beslutningsvariablene $x_{i,t,k}$ blot fortæller, hvornår toget afgår fra depot k . Togene bliver altså ikke ”fulgt rundt” som i model 2.

En anden faktor, der øger antallet af begrænsninger i model 1, er, at det ved hjælp af tre sæt begrænsninger sikres, at der afgår cirka lige mange tog i hver retning. Dette krav er ikke nødvendigvis opfyldt i model 2, hvor der altså i princippet godt kan indsættes f.eks. 2 tog fra København Nord og ingen fra København S. I mange tilfælde vil model 2 dog give en optimal løsning, der netop overholder dette krav; hvis man for eksempel skal indsætte 6 tog fra København, vil det selvfølgelig være hurtigst at indsætte 3 i hver retning. Hvis vi derimod har 3 tog på Ballerup, 2 på København og 1 på Klampenborg, vil begge togene fra København blive sendt i nordgående retning mod Klampenborg i en optimal (hurtigste) genindsættelse.

En væsentlig forskel i modellerne er selvfølgelig, hvordan de har valgt at overholde de fire krav fra afsnit 3.3:

Lokomotivførere på depoterne

Model 1 sikrer dette direkte vha. begrænsning (16), mens dette i model 2 sikres indirekte ved hjælp af vores A -matrix. Dette kan for eksempel ses ved, at vi i Ballerup først kan indsætte tog fra og med tidsenhed t_3 , da der ellers ikke vil kunne nå at komme lokomotivførere derud.

Alle udtagne tog genindsættes

I model 1 sikrer begrænsningerne (9) og (10), at alle udtagne tog på de forskellige depoter også bliver genindsat. I model 2 sikres dette af (25). Forskellen består her i, at model 1 ser på hver endestation og gennemkørende depot for sig, hvorimod model 2 ser på det totale antal over alle depoter.

Ét tog per togstreng

I model 1 sikres det i begrænsning (7). I model 2 sikres dette krav i begrænsningerne (24) og (25).

Tog indsættes sekventielt

Model 2 sikres dette ved hjælp af begrænsningerne (23) og (24). Her sikres det altså, at der altid kommer et tog bagefter et, som lige er blevet indsat. I model 1 sikres det gennem (15), at togene indsættes sekventielt fra de enkelte retningsdepoter. Dog ligger det kun implicit (og som en antagelse) i modellen, at eftersom vi minimerer $maxtognummer$, vil modellen selv sørge for at få togene ”skubbet sammen”, så de indsættes sekventielt fra de enkelte depoter.

Som et modeksempel på antagelsen om, at model 1 indsætter tog sekventielt, kan vi se på løsningen 321, altså at der skal indsættes 3 tog fra Klampenborg, 2 fra København H og 1 fra Ballerup. Her vælger modellen at indsætte Klampenborg-togene i tidsenhederne 3,4,5, København H-togene i tidsenhed 3 (sydgående) og 5 (nordgående), samt Ballerup-toget i tidsenhed 3. Denne løsning er ikke sekventiel i Ballerup, da der ikke vil afgå noget tog i tidsenhed 4 (KHS-toget når først frem i tidsenhed 5).

Med hensyn til uddata, så giver model 1 os direkte de tognumre, vi har brug for. Model 2 giver i GAMS-outputtet de søjler, der er blevet anvendt i løsningen. Man kan derefter hurtigt i A -matricen se, hvordan togene skal indsættes. Hermed vil man også hurtigt kunne udregne de tognumre, som planlæggerne skal bruge, samt de tog, der skal transportere lokomotivførere ud til de ventende tog på depoterne.⁵

Objektfunktionens værdi i de to modeller er desuden meget forskellig. I model 1 får vi de to sidste cifre i tognummeret på det tog, som passerer København H sidst, efter det er blevet genindsat. Herudfra kan man hurtigt se, hvornår hele genindsættelsen vil være overstået. I model 2 siger objektfunktionsværdien ikke rigtig noget, men bruges blot som faktor til at minimere efter.

En sidste bemærkning i dette afsnit skal gå til selve størrelsen af modellerne. De to GAMS-implementationer (som er nærmere beskrevet i afsnit 5) genererer modeller, som er meget forskellige i omfang. Model 2 kan for eksempel løses ved hjælp af test-licensen til GAMS⁶, hvorimod model 1 indeholder alt for mange søjler og begrænsninger til at kunne dette⁷.

⁵Hos DSB S-tog bliver dette gjort ved at indtaste modellens output i et Excel-ark, der beregner de ønskede tognumre og hvilke tog, der skal transportere lokomotivførerne ud til depoterne.

⁶Der genereres — 48 rows 25 columns 209 non-zeroes — 24 discrete-columns

⁷Her genereres der — 768 rows 300 columns 1,964 non-zeroes — 299 discrete-columns

4. Videreudvikling af modeller

Vi skal i dette afsnit se nærmere på de muligheder, der er for udvidelse af ovenstående to modeller. Nogle af ideerne bygger på forslag fra [3], s.19f. Med udvidelser definerer vi ændringer af modellerne, der fører til forbedringer eller ny funktionalitet. Som det vil kunne ses i afsnit 5.2, er der ikke den store grund til at fokusere på tiltag, der reducerer køretiden for modellerne, da denne for begge modellens vedkommende er under 5 CPU-sekunder. Generelt henvises der til afsnit 5.2 for resultater af de foreslåede udvidelser, samt sammenligning af oprindelige modeller.

Inden vi påbegynder selve videreudviklingen, vil vi starte med at bemærke, at begrænsningerne (8) i model 1 er redundante. Dette følger af den måde, hvorpå konstanterne $in_{i,t,k}$ er konstrueret. $in_{i,t,k}$ sikrer netop, at der ikke kan indsættes to tog i samme togstreng, altså ikke indsættes to tog i samme tidsenhed fra samme depot. Da beslutningsvariablene $x_{i,t,k}$ er bundet til $in_{i,t,k}$ gennem begrænsningerne (14), vil (8) trivielt altid være opfyldt.

Vi skal nu se på forskellige udvidelsesmuligheder for de to modeller, og vi starter med en af forskellene i modellerne.

4.1. Bløde krav

Vi vil nu se på det ”bløde” krav, der siger, at der skal indsættes lige mange tog i hver retning fra de gennemkørende depoter. For model 1 er dette et eksplicit krav, der er udmøntet i begrænsningerne (11), (12) og (13). I model 2 stilles dette ikke som krav. Kravet udspringer af gammel praksis på DSB S-tog, og dets relevans og berettigelse giver anledning til undren. I vores øjne bør en optimal indsættelse ikke tage hensyn til dette krav men snarere tage hensyn til, hvornår de enkelte tog bliver indsat i normal drift igen. Vi mener derfor, at disse krav bør udelades fra en genindsættelsesmodel, som det er tilfældet i model 2. Vi har dog implementeret kravene i denne model også for at sammenligne løsningernes kvalitet og dermed disse kravs betydning. Ligeledes har vi fjernet kravet fra model 1 (se Model 1* og Model 2* i afsnit 5.2).

I model 1 er der blot fjernet begrænsningerne (11), (12) og (13). I model 2 indfører vi en ny mængde $J^I \subset J$ indeholdende de søjler, der har med indsættelser på de gennemkørende depoter at gøre. Ser vi på linie C og dennes A-matrix fra afsnit 3.5, får vi, at $J^I = \{7, \dots, 12, 19, \dots, 24\}$. Desuden introducerer vi tilsvarende variablene d_k^I , som angiver antallet af tog indsat i hver retning fra de gennemkørende depoter. Vi indfører nu begrænsninger, der svarer til de nævnte fra model 1:

$$\sum_{j \in J^I} x_j = d_k^I, \quad \forall k \in K^I \quad (26)$$

$$d_k^I \geq \left\lfloor \frac{d_k}{2} \right\rfloor, \quad \forall k \in K^I \quad (27)$$

$$d_k^I \leq \left\lceil \frac{d_k}{2} \right\rceil, \quad \forall k \in K^I \quad (28)$$

4.2. Alternative objektfunktioner

Den nuværende objektfunktion i model 1 minimerer de to sidste cifre i højeste tognummer brugt i genindsættelsen, regnet fra passage af København H. Dette er i vores øjne ikke nogen god indikator for, hvor god en genindsættelse er, eller hvor tidligt den er afsluttet (som i sidste ende er det ultimative mål). Denne formulering tager kun højde for den sidste indsættelse af et tog og ikke for de resterende indsættelser på linien. Hvis denne sidste indsættelse sker fra et endedepot, vil objektværdien blive unødvendigt stor, da der kræves ekstra tid, før dette tog når ind og passerer København H.

I model 2 ser objektfunktionen ganske anderledes ud, idet der her minimeres den samlede indsættelsestid for alle tog på linien. Ulempen er her, at man ikke umiddelbart minimerer efter den sidste indsættelse, men efter de samlede indsættelser.

Som en bedre objektfunktion foreslår vi derfor at minimere først over sidste indsættelse og derefter samlet indsættelsestid. Konkret giver dette i model 1 følgende revideres objektfunktion:

$$z = \min N \cdot \text{MaxIndsaet} + \sum_{i,t,k} (x_{i,t,k} \cdot t) \quad (29)$$

hvor konstanten N er valgt tilpas stor.⁸ Her er MaxIndsaet valgt som vores sidste indsættelse ved hjælp af følgende sæt begrænsninger:

$$\text{MaxIndsaet} \geq x_{i,t,k} \cdot t, \quad \forall i \in I, t \in T, k \in K \quad (30)$$

For model 2s vedkommende er ændringerne næsten identiske:

$$z = \min N \cdot \text{MaxIndsaet} + \sum_j (x_j \cdot c_j) \quad (31)$$

hvor konstanten N er valgt tilpas stor. MaxIndsaet er igen valgt som vores sidste indsættelse ved hjælp af følgende sæt begrænsninger:

$$\text{MaxIndsaet} \geq c_j \cdot x_j, \quad \forall j \in J \quad (32)$$

Resultaterne kan ses i afsnit 5.2 (se Model 1** og Model 2**).

4.3. Ekstra disponible tog på depoterne

I dette afsnits modeller er kravet om lige mange tog i begge retninger udeladt, som beskrevet ovenfor. Vores sidste videreudvikling bygger nu på, at der potentielt kan stå et eller flere ekstra tog på depoterne i forvejen, som måske med fordel ville kunne bruges i genindsættelsen. Med ekstra tog menes tog, som ikke er i brug på nuværende tidspunkt og som derfor umiddelbart vil kunne indsættes på den linie, som genindsættelsen skal foregå på.

I model 1 løses dette problem ved at opbløde begrænsningerne (9) og (10) fra hårde til bløde lighedstegn ($\leq$), samt ved at ændre inddata til modellen, så d_k fortæller, hvor mange tog vi *kan* indsætte fra de forskellige depoter. Til sidst tilføjer vi en konstant

⁸For linie C's vedkommende er $N = 100$.

W , som angiver, hvor mange togstreng, der kører på linien. Dette giver anledning til følgende begrænsning:

$$\sum_{i,t,k} x_{i,t,k} = W \quad (33)$$

I model 2 gør vi næsten det samme, dog med en begrænsning mindre. Vi bløder på samme måde op for begrænsningerne (25) ($\leq$), lader d_k angive antallet af tog til rådighed på depot k , indfører konstanten W og tilføjer så følgende begrænsning:

$$\sum_j x_j = W. \quad (34)$$

Som allerede nævnt kan testresultater ses i afsnit 5.2.

4.4. Fremtidig videreudvikling

Tre videreudviklinger har vi ikke kunnet nå at se på i dette projekt:

Straf i objektfunktionen

Som beskrevet ovenfor mener vi ikke, at kravet om lige mange tog i hver retning fra gennemkørende depoter er et ultimativt (hårdt) krav. Man kunne dog vælge at se på muligheden for at implementere kravet i objektfunktionen, så denne kunne tage stilling til to ellers ens løsninger, der afviger i netop dette krav. På denne måde vil objektfunktionen vælge efter 3 kriterier: først sidste indsættelse, så samlet indsættelsestid, og til sidst efter antallet af depoter, hvor kravet om lige mange tog i hver retning er opfyldt. Endnu mere finkornet kunne man gøre det ved at se, i hvor høj grad kravet er opfyldt på de enkelte depoter. Andre ændringer i objektfunktionen i form af tilføjelse af ekstra bløde krav kunne ligeledes sagtens være mulige.

Genindsættelse af flere linier

Modellerne kunne for så vidt udvides til at håndtere genindsættelse af flere udtagne linier på én gang. Dette vil være specielt relevant for linier, som fletter, altså hvor et tog kører skiftevis på den ene og den anden linie. Komplexiteten af denne udvidelse vil dog være af betragtelig størrelse.

Lade tog køre længere inden udtagelse

En sidste mulighed er at lade togene køre længere end krævet, inden de udtages. Her vil der potentielt kunne opnås væsentlig hurtigere genindsættelser, specielt hvis man udtager flere tog på gennemkørende depoter; særligt på København H, hvor mandskabsdepotet er placeret. I vores øjne er dette dog ikke en faktor, som en genindsættelsesmodel vil kunne indeholde direkte. Det vil blot give et andet input til modellen om, hvor togene er placeret, ligesom starttidspunktet for genindsættelsen vil blive forsinket. Desuden vil det at lade togene køre længere sandsynligvis skabe mere trafik omkring København H, hvilket man netop forsøger at undgå ved at udtage toglinier. Dette er kun gældende for det gennemkørende depot København H. Andre gennemkørende depoter vil man fra endepoter sandsynligvis med fordel kunne lade tog køre videre til, da vi for det første kan få lokomotivførere derud hurtigere, og for det andet har mulighed for at genindsætte tog i flere retninger.

5. Implementation og test

I dette afsnit skal vi se nærmere på, hvordan modellerne klarer sig over for hinanden. Alle modeller er testet på data om linie C i S-togsnettet, der kører mellem Klampenborg og Ballerup, over København H. Først skal vi i afsnit 5.1 beskrive de forskellige modellers implementationer og forskelle (kildekoden kan ses i Appendix B). Sidst, men ikke mindst, skal vi se, hvordan de forskellige modeller klarer sig mod hinanden i forhold til forskellige parametre. Mere herom i afsnit 5.2.

5.1. Implementation af modeller i GAMS

Alle modeller og disses videreudviklinger præsenteret i afsnit 3.3 og 4 er implementeret i GAMS.⁹ GAMS er et modelleringssprog, der anvender ILOGs CPLEX¹⁰ som ”motor”. GAMS er i stand til at modellere et utal af forskellige IP-, MIP- og BIP-modeller, samt en del mere specialiserede matematiske modeller. GAMS er blevet valgt som implementeringssprog af flere grunde: model 2 fik vi udleveret kildeteksten i GAMS til (da GAMS anvendes i DSB S-tog til løsning af matematiske modeller), og det ville derfor være mest sammenligneligt at implementere de yderligere modeller i dette også. Desuden havde en af os en smule erfaring med GAMS i forvejen, ligesom det er givtigt for forståelsen af virksomheden at anvende de samme redskaber og metoder, som finder anvendelse i erhvervslivet.

For alle modellers vedkommende er der lavet to filer: en med selve implementationen (”modelX.gms”), og en med de nødvendige inddata til modellen (eksempelvis ”C.txt”). Modellerne er så vidt muligt implementeret, så de kan modtage inddata for samtlige 7 forskellige linier i S-togsnettet. Det er altså som udgangspunkt kun nødvendigt at tilrette .txt-filerne for hver linie. Grundet størrelsen af modellerne samt for bedre at kunne sammenligne køretider er de alle afviklet og testet på IMF’s GAMS-licens.¹¹ Vi skal nu kort se på nogle tekniske udfordringer i implementationen, samt fremhæve centrale dele.

I GAMS anvendes der overordnet set 3 slags datastrukturer til at beskrive inddata:

SETS

SETS er statiske mængder af forskellige typer. De kan indeholde talværdier (som f.eks. indeksemængder) eller navne (som f.eks. navne på depoter). En væsentlig ting at bemærke i forhold til GAMS’ brug af mængder er, at man kan lave delmængder af mængder. Vi har dog ikke i GAMS umiddelbart kunne finde en mulighed for eksempelvis at lade begrænsninger løbe over udvalgte mængder i et SET uden specifikt at lave et ekstra del-SET til de pågældende elementer. Dette kan f.eks. ses i implementationen af model 1’s begrænsninger (11), (12) og (13).

SCALARS

SCALARS beskriver de faste, ikke-indekserede konstanter, der anvendes i modellen. I vores tilfælde er der f.eks. brugt SCALARS til konstanten *initialtog*.

⁹GAMS: *General Algebraic Modelling System* (se evt. <http://www.gams.com/>)

¹⁰CPLEX er et samling af løsningsværktøjer til matematisk programmering og modellering, som mange forskellige programmerings- og modelleringssprog kan udnytte, heriblandt GAMS. For mere information om CPLEX, se <http://www.ilog.com/products/cplex/>.

¹¹Model 2 samt dennes videreudviklinger kan dog faktisk køres på en gratis testlicens af GAMS.

PARAMETERS

PARAMETERS er (modsat SCALARS) *indekserede* konstanter; altså værdier der tilknyttes alle elementer i et SET eller krydsprodukter af SETS. Et eksempel fra model 1 er parameteren `clock(k)`, der for hvert depot angiver, hvor mange tidsenheder det tager en lokomotivfører at nå derud.

Der er udover ovennævnte tre også muligheder for TABLES (flerdimensionelle parametre) og andre datastrukturer, men dem har vi ikke fundet anvendelse for i dette projekt.

Det er selvfølgelig også muligt at definere forskellige typer af variable og grænser for samme. I model 1 er der f.eks. defineret heltalsvariablene `DIK(KI)`, `start(k)` og `slut(k)`, som løber over mængden af hhv. gennemkørende depoter (KI) og depoter (k). En væsentlig ting at bemærke ved GAMS er, at heltalsvariable som standard vil få sat en (for os ukendt) lav øvre grænse, hvilket har givet problemer undervejs i fejlfindingen. Derfor er unødigt mange variable i implementationerne angivet som frie for at komme dette problem til livs.

Herudover bliver begrænsninger defineret som EQUATIONS, og der defineres en MODEL, der indeholder alle begrænsninger. Til sidst løses modellen vha. Mixed Integer Programming-rutiner (MIP). ([5], s.4ff)

Model 1 er implementeret fra bunden ud fra gennemgangen i afsnit 3.4. Som man kan se i kildekoden i Appendix B, er det lavet flere mængder af depoter end beskrevet i den matematiske model. Det har været nødvendigt at lave hele fire mængder af depoter: de tre depoter overordnet (D), retningsdepoterne (k), endedepoterne (KE) samt de gennemkørende depoter (KI). Her ville det have været en fordel at kunne have nøjes med én mængde af depoter, som man undervejs kunne vælge elementer ud fra. Udover disse betragtninger er implementationen ligetil og følger begrænsningerne i modellen.

Model 2 er væsentlig anderledes i dens opbygning. Da mange af begrænsningerne i model 2 ligger implicit i inddata til modellen, er datafilen til model 2 væsentlig længere end for model 1's vedkommende. Til gengæld er selv modellen simplere. Vi har fået udleveret kildekoden til model 2 fra DSB S-tog, så det er denne i mestendels uforandret form, vi har gengivet i Appendix B. Koden er blevet ryddet en smule op og relateret til gennemgangen af den matematiske model i afsnit 3.5. Det skal bemærkes, at `stamme(j)`-parameteren i model 2 principielt er overflødig, da den kun optræder i output. Værd at bemærke er desuden, at implementationen af denne model faktisk beregner genindsættelser for *alle* mulige paceringer af de 6 tog på linie C. Vi har for nemheds skyld under testen også overført dette til model 1 og dennes videreudviklinger, men undladt det i kildekoden.

Vi vil nu ganske kort gennemgå de ændringer, der er foretaget i modellerne i forbindelse med videreudvikling (se afsnit 4) og test. Kilkoden er ikke vedlagt, da der er tale om meget små ændringer eller tilføjelser. Først ser vi på ændrede objektfunktioner. I model 1 er (29) implementeret ved at oprette en ny fri variabel `MaxIndsaet`, som lagrer sidst indsatte tog ud fra den nye begrænsning (30). Desuden ændrer vi den oprindelige objektfunktion `objective`:

EQUATIONS

<code>MaxInd(i,t,k)</code>	Bruges til at finde sidste indsaettelse
<code>objective</code>	Objektfunktionen (minimer maxtognummer)

```

;
MaxInd(i,t,k)..   MaxIndsaet =G= x(i,t,k)*ORD(t);
objective..       z =E= 100*MaxIndsaet + sum((i,t,k),x(i,t,k)*ORD(t));

```

For model 2's vedkommende ser ændringer næsten ens ud:

```

EQUATIONS
    Objective
    MaxIndsaettelse(j)
;
    Objective..       z =E= 100*MaxIndsaet + sum(j,c(j)*x(j));
    MaxIndsaettelse(j)..   MaxIndsaet =G= c(j)*x(j);

```

For at implementere det bløde krav om lige mange tog i hvert retning i model 2 (altså begrænsningerne (26), (27) og (28)), indfører vi først et SET `KI(depotReinsertionDirections) /KHS, KHN/` indeholdende retningsdepoter på nær endestationer, samt variablene `DIK(KI)`; meget ala model 1. Herudover er vi dog også nødt til at indføre parameteren `aa(rd,t,j)`, som er en delmatrix af A -matricen. `aa(rd,t,j)` indeholder de søjler fra A , der har med de retningsdepoter, som ikke er endedepoter, at gøre. For linie C vil det sige KHN (søjle 7-12) og KHS (søjle 19-24). De relevante data hives automatisk ud af A -matricen med følgende kodelump:

```

SETS
    KHNslots(columns) / 7*12 /
    KHSslots(columns) / 19*24 /
;
LOOP((t,KHNslots),
    aa("KHN",t,KHNslots) = a("KHN",t,KHNslots);
);
LOOP((t,KHSslots),
    aa("KHS",t,KHSslots) = a("KHS",t,KHSslots);
);

```

Herefter tilføjer vi ovennævnte begrænsninger på følgende måde:

```

EQUATIONS
    TwoWay1(KI)          % (30)
    TwoWay2(KI)          % (31)
    TwoWay3(KI)          % (32)
;
    TwoWay1(KI)..        sum((j,t),aa(KI,t,j) * x(j)) =E= DIK(KI);
    TwoWay2(KI)..        DIK(KI) =G= FLOOR(insertIn("DepotKH")/2);
    TwoWay3(KI)..        DIK(KI) =L= CEIL(insertIn("DepotKH")/2);

```

For at implementere muligheden for at have ekstra tog stående på depoterne, har vi i model 1 opblødt begrænsningerne (9) og (10) til bløde lighedstegn ($\leq$), ladet `DA(k)` i stedet angive hvor mange tog, der er til rådighed på hvert depot, indført SCALAR'en `IALT = 6` (for at angive det antal, der skal indsættes på linien), samt indført følgende begrænsning:

```

EQUATIONS
    TotalReinserted      "Det totale antal genindsatte tog matcher antal togstreng paa linien";
    TotalReinserted..    sum((i,t,k),x(i,t,k)) =E= IALT;

```

For model 2's vedkommende gør vi præcis det samme, bortset fra, at det nu er begrænsning (25), vi bløder op:

```

EQUATIONS
    TotalReinserted      "Det totale antal genindsatte tog matcher antal togstreng paa linien";
    TotalReinserted..    sum(j,x(j)) =E= IALT;

```

5.2. Testresultater

Vi har, som nævnt i afsnit 4, testet de forskellige modeller og videreudviklinger. Vi har valgt at tage udgangspunkt i linie C i vores test (se Figur 6 på side 26). Linie C har 3 depoter, hvor det ene er det gennemkørende depot København H. Linien består af 6 togstrengene, hvilket giver 28 muligheder¹² for, hvor togene er ved udtagelsestidspunktet. Mange af disse muligheder er teoretiske, da det med meget ringe sandsynlighed vil ske, at alle 6 tog på linien udtages i f.eks. Klampenborg.

Ud over model 1 og model 2, testede vi model 1*, som var model 1 uden kravet om lige mange tog i hver retning og model 2*, som var model 2 med krav om lige mange tog i hver retning. Derudover testede vi model 1**, som var model 1* med ny objektfunktion og model 2**, som var model 2 med ny objektfunktion (som beskrevet i afsnit 4.2)

Da de to oprindelige modeller har en meget forskellig objektfunktion, vælger vi at teste modellerne efter flere forskellige kriterier. Først undersøger vi, hvorvidt modellerne kan løses til optimalitet for alle 28 instanser (*OPT*). De næste to kriterier er de to oprindelige objektfunktioner (*OM1* og *OM2*).

Som et tredje kriterie har vi valgt at se på, hvornår sidste indsættelse finder sted, uafhængigt af hvor dette tog bliver indsat (*SI*). Sidste indsættelse blev vurderet i forhold til model 1** og model 2**, som specifikt minimerer dette kriterie. Til slut undersøgte vi, hvor mange af løsningerne, der opfyldte kravet om sekventialitet (*SK*: togene kommer fortløbende efter indsættelse), at der blev sendt lige mange tog i hver retning (*2V*) og noterede den samlede køretid for alle 28 instanser i CPU-sekunder (*TID*). Resultaterne kan findes i Appendix A, og er sammenfattet i Tabel 2 herunder:

	Model 1	Model 2	Model 1*	Model 2*	Model 1**	Model 2**
OPT	28	28	28	28	28	28
OM1	28	28	28	28	28	28
OM2	14	28	25	20	28	28
SI	24	28	25	28	28	28
SK	20	28	26	28	26	28
2V	28	20	20	28	20	20
TID	3,4	1,2	3,2	1,3	3,0	1,3

Tabel 2: Testresultater (grupperede) for linie C.

Ved vores test blev alle modeller løst til optimalitet indenfor en meget overskuelig tid. Alle modeller kan altså bruges til at løse genindsættelsesproblemet i realtid. Alle modeller fik de optimale løsninger i forhold til model 1 kriteriet, ved alle 28 mulige indsættelser. Set i forhold til model 1*, 1** og 2** er dette ikke overraskende, da sidste indsættelse typisk sker på endedepoterne, som også er der, det højeste tognummer vil forekomme. At højeste tognummer i model 2 og model 2* bliver lige så lavt som i model 1, er derimod lidt overraskende. En forklaring på dette kan dels findes i kravet om sekventialitet, og

¹²Det giver 28 via følgende udregning: Ved indsættelse af ingen tog i Klampenborg er der 7 muligheder for København H (0-6 tog) og derefter er Ballerup fastsat, ved 1 tog i Klampenborg er der 6 muligheder i København H osv. Det giver i alt $7 + 6 + 5 + 4 + 3 + 2 + 1 = 28$ muligheder

dels at det er den samlede indsættelsestid, der minimeres.

Vi testede herefter de forskellige modeller efter *OM2* kriteriet. Her havde model 1 i halvdelen af tilfældene en samlet indsættelsestid, der var langsommere end model 2, hvilket ikke er overraskende, da modellen blot minimerer sidste tognummer. Hvis vi så fjernede kravet om lige mange tog i begge retninger (1^*), så blev der yderligere 6 løsninger, der var optimale i forhold til *OM2*-kriteriet. I model 2^* , hvor vi tilføjede den begrænsning, at der skulle lige mange tog i begge retninger, forringede dette objektværdien i 8 tilfælde. De to modeller med ny objektfunktion får ikke overraskende optimale løsninger i alle tilfælde.

Ved testkriteriet *SI* får næsten alle modeller samme løsning, kun model 1 og model 1^* har nogle løsninger med senere sidste indsættelse. Disse sene indsættelser fremkommer alle, når der indsættes 2 eller flere tog fra København H, hvilket igen skyldes at det blot er tognummeret, der minimeres, altså hvornår sidste indsættelse passerer København H. Vi kiggede herefter kort på en lidt større toglinie C^+ , som er linie C med det ekstra endepot Frederikssund. At der kommer et ekstra depot, gør samtidig at Ballerup bliver et gennemkørende depot. Denne større linie ændrede i midlertid ikke ved at model 2 fik ligeså hurtige indsættelser som model 2^{**} . Om dette altid vil gælde, har vi ikke haft tid til at undersøge.

Hvis man ser på *SK*, kan vi se, at model 1 og model 1^* ikke opfylder det hårde krav om, at togene kommer fortløbende efter indsættelse. Antallet af løsninger, der ikke opfylder *2V*-kriteriet, er begrænset. For modellerne uden krav om, at der skal lige mange tog i hver retning, har vi i 20 ud af de 28 tilfælde togindsættelser, der opfylder kravet. Dette antal, vil man igen kunne forvente, ville blive øget, hvis man så på en større toglinie, idet der ville være flere gennemkørende depoter.

Vi valgte at videreudvikle model 1^{**} og model 2^{**} , så de kan tage evt. ekstra tog, der holder på depoterne med i modellen (model $1v$ og $2v$). Hvis man igen skulle vurdere på samtlige løsninger på linie C, ville man skulle se på 287¹³ instanser.

Vi har valgt at teste de to videreudviklede modeller i tre mulige indsættelser, da alle 287 mulige vil minde meget om hinanden (løsningerne til de to modeller vil være blandt de oprindelige 28 indsættelser). Vi tager udgangspunkt i den situation, hvor der står et ekstra tog på hvert af depoterne, altså et ekstra tog på Ballerup, Klampenborg og København H. Under disse forudsætninger har vi set på tre tilfælde: hvor der trækkes 1 tog ud i Klampenborg, 2 tog ud på København H og 3 tog ud i Ballerup; hvor der

¹³Vi kommer frem til 287 ved følgende udregning: på hver station kan der teoretisk indsættes mellem 0 og 6 tog ved hver indsættelse, altså $7^3 = 343$. Mange af disse 343 indsættelser opfylder ikke, at det samlede antal tog er mindst 6. Vi skal altså finde de muligheder, hvor det samlede antal tog er færre end 6. Vi starter med at se på KL og tilfældet, hvor der ikke står nogen tog. Da vil der på KH kunne indsættes 0-5 tog, hvor det samlede antal stadig vil kunne være mindre end 6. Hvis der blev indsat 0 fra KH, ville der være 6 muligheder fra BA; 5 fra BA med 1 fra KH osv. Dette giver $6 + 5 + 4 + 3 + 2 + 1 = 21$ muligheder, hvis der ikke indsættes nogen tog fra KL. Indsættes der 1 tog fra KL, vil vi kunne indsætte 0 – 4 tog fra KH og 4 – 0 tog fra BA. Ved samme overvejelser som i tilfældet før, giver dette $5 + 4 + 3 + 2 + 1 = 15$. Ved 2-6 tog fra KL giver det 10, 6, 3, 1 og 0 indsættelser, der ikke kan lade sig gøre. I alt får vi $7^3 - 21 - 15 - 10 - 6 - 3 - 1 = 287$ mulige indsættelser.

trækkes tre tog ud på Klampenborg og København H; og det nok mest realistiske, hvor der trækkes 4 tog ud på København H, 1 tog i Klampenborg og 1 tog i Ballerup.

Vi har testet løsningerne overfor de oprindelige modeller 1 og 2, samt overfor $1^{**}/2^{**}$ (da disse havde samme løsning). Vi har opskrevet løsningerne på model 1 og 2, så de står på samme form som model 1^{**} og 2^{**} , hvor det første ciffer angiver sidste indsættelsestidspunkt og de to efterfølgende den samlede indsættelsestid. *TFD* angiver fordelinger af tog på de forskellige depoter (Klampenborg-København-Ballerup), hvor de videreudviklede modeller i alle tilfælde har et ekstra tog på hvert depot, de kan sætte ind. Resultaterne af de forskellige tests er samlet i Tabel 3.

Ved den første test gav model $1v$ resultatet, at der skulle indsættes 2 tog fra KHS, 1 fra KHN, 2 fra KL og 1 fra BA. Model $2v$ fik en anden løsning end model 1, men fik samme værdi i objektfunktionen. Model $2v$ fik løsningen 1 tog fra KHS, 2 fra KHN, 1 fra KL og 2 fra BA i test 1. I test 2 og 3 fik de to videreudviklede modeller samme løsning: 1 tog fra BA, 2 tog fra KHN og KHS og et tog fra KL, og 2 tog fra KHS, 3 fra KHN og 1 fra BA.

Test	TFD	Model 1	Model 2	Model $1^{**}/2^{**}$	Model $1v$	Model $2v$
1	123	623	520	520	418	418
2	330	524	518	518	316	316
3	141	316	316	316	314	314
TID	-	0,35	0,18	0,31 / 0,17	0,37	0,208

Tabel 3: Testresultater for videreudviklede modeller (linie C)

Ved alle 3 test tager de videreudviklede modeller ikke overraskende det ekstra tog, der står på København H, som en del af løsningen. At de videreudviklede modeller giver bedre løsninger i alle tilfælde, er derfor heller ikke særlig overraskende. At det kan betale sig at tage ekstra tog i betragtning, illustrerer test 1 og 2 tydeligt. I test 2 er de videreudviklede modeller markant bedre end de øvrige. Sidste indsættelse er 40 min. hurtigere og samlet indsættelsestid 40 min. hurtigere end for model 2 og $1^{**}/2^{**}$ (model 1 er endnu dårligere). Tiden er den samlede køretid for alle 3 test, men som man kan se er tiden stadig overskuelig og modellerne kan løses i realtid.

6. Konklusion

Med hensyn til DSB S-tog kan vi på baggrund af dette projekt konkludere, at der findes en sektion, der arbejder med problemstillinger, som ligger direkte i forlængelse af de kurser, vi har haft inden for emnet på Københavns Universitet. Denne sektion har ligeledes berettiget sin eksistens overfor ledelsen, idet dens resultater inden for operationsanalyse har finansieret både den egentlige eksistens samt løbende udvidelse af sektionen.

På det faglige plan har vi sammenlignet de to modeller stillet til rådighed af DSB S-tog og konkluderet, at model 1 ikke opfylder det hårde krav om sekventialitet i genindsættelsen, og at model 2 giver optimale løsninger på alle centrale parametre, som modellerne er blevet testet efter. Vi har set, at man med fordel kan tage ekstra tilgængelige tog med ind i modellen for at opnå bedre løsninger, hvis forbedringsgrad dog er situationsbestemt. På trods af, at genindsættelsesproblemet er en forholdsvis virksomhedsspecifik og begrænset problemstilling, er der stadig flere muligheder for videre udvikling og arbejde med modellerne og disses forudsætninger.

Litteratur

- [1] DSB S-tog:
<http://www.dsb.dk/Om-DSB/Virksomheden/Organisationen/DSB-S-tog/> (2009).
- [2] J.J. Groth m.fl.: "*Disruption Management in Passenger Railway Transportation*", i *Econometric Institute Report EI2007-05*, Econometric Institute, Erasmus University Rotterdam (2007).
- [3] J.J. Groth, J. Clausen, J. Larsen: "*Optimal Resinsertion of Cancelled Train Line*", i *Technical Report IMM-Technical Report-2006-13*, Informatics & Mathematical Modelling, Danmarks Tekniske Universitet (2006).
- [4] N.J. Rezanova: "*The Train Driver Recovery Problem – Solutions Method and Decision Support System Framework*", endnu ikke publiceret, Informatics & Mathematical Modelling, Danmarks Tekniske Universitet (2009)
- [5] Rosenthal, R.E.: "*A GAMS Tutorial*:"
<http://www.gams.com/dd/docs/gams/Tutorial.pdf>.
- [6] L. A. Wolsey: "*Integer Programming*", John Wiley & Sons, Inc. (1998).

A. Testresultater for Linie C

TFD	Model 1				Model 2				Model 1*				Model 2*				Model 1**		Model 2**	
	OM1	OM2	SI	SK	OM1	OM2	SI	2V	OM1	OM2	SI	SK/2V	OM2	SI	OMNy	SK/2V	OMNy	SK/2V	OMNy	2V
006	34	33	8	Y	34	33	8	Y	34	33	8	Y/Y	33	8	833	Y/Y	833	Y/Y	833	Y
015	33	28	7	Y	33	28	7	Y	33	28	7	Y/Y	28	7	728	Y/Y	728	Y/Y	728	Y
024	32	26	6	N	32	23	6	N	32	23	6	Y/N	26	6	623	Y/N	623	Y/N	623	N
033	31	21	6	N	31	18	5	N	31	18	5	Y/N	21	5	518	Y/N	518	Y/N	518	N
042	30	19	4	Y	30	16	4	N	30	16	4	Y/N	19	4	416	Y/N	416	Y/N	416	N
051	29	17	4	Y	29	14	3	Y	29	17	4	Y/Y	14	3	314	Y/Y	314	Y/Y	314	Y
060	28	12	3	Y	28	12	3	Y	28	12	3	Y/Y	12	3	312	Y/Y	312	Y/Y	312	Y
105	33	30	7	Y	33	30	7	Y	33	30	7	Y/Y	30	7	730	Y/Y	730	Y/Y	730	Y
114	32	25	6	N	32	25	6	Y	32	25	6	N/Y	25	6	625	N/Y	625	N/Y	625	Y
123	31	23	6	N	31	20	5	N	31	20	5	N/N	23	5	520	Y/N	520	Y/N	520	N
132	30	21	4	Y	30	18	4	Y	30	18	4	Y/Y	18	4	418	Y/Y	418	Y/Y	418	Y
141	29	16	3	Y	29	16	3	Y	29	16	3	Y/Y	16	3	316	Y/Y	316	Y/Y	316	Y
150	29	17	4	Y	29	14	3	Y	29	14	3	Y/Y	14	3	314	Y/Y	314	Y/Y	314	Y
204	32	27	6	Y	32	27	6	Y	32	27	6	Y/Y	27	6	627	Y/Y	627	Y/Y	627	Y
213	31	25	5	Y	31	22	5	Y	31	22	5	Y/Y	22	5	522	Y/Y	522	Y/Y	522	Y
222	30	20	4	Y	30	20	4	Y	30	20	4	Y/Y	20	4	420	Y/Y	420	Y/Y	420	Y
231	30	21	4	Y	30	18	4	Y	30	18	4	Y/Y	18	4	418	Y/Y	418	Y/Y	418	Y
240	30	19	4	Y	30	16	4	N	30	16	4	Y/N	19	4	416	Y/N	416	Y/N	416	N
303	31	24	5	Y	31	24	5	Y	31	24	5	Y/Y	24	5	524	Y/Y	524	Y/Y	524	Y
312	31	25	5	N	31	22	5	Y	31	25	6	Y/Y	22	5	522	Y/Y	522	Y/Y	522	Y
321	31	23	5	N	31	20	5	N	31	26	6	Y/N	23	5	520	Y/N	520	Y/N	520	N
330	31	24	5	Y	31	18	5	N	31	18	5	Y/N	21	5	518	Y/N	518	Y/N	518	N
402	32	27	6	Y	32	27	6	Y	32	27	6	Y/Y	27	6	627	Y/Y	627	Y/Y	627	Y
411	32	25	6	N	32	25	6	Y	32	25	6	Y/Y	25	6	625	N/Y	625	N/Y	625	Y
420	32	26	6	N	32	23	6	N	32	23	6	Y/N	26	6	623	Y/N	623	Y/N	623	N
501	33	30	7	Y	33	30	7	Y	33	30	7	Y/Y	30	7	730	Y/Y	730	Y/Y	730	Y
510	33	28	7	Y	33	28	7	Y	33	28	7	Y/Y	28	7	728	Y/Y	728	Y/Y	728	Y
600	34	33	8	Y	34	33	8	Y	34	33	8	Y/Y	33	8	833	Y/Y	833	Y/Y	833	Y

TFD: Mulige fordelinger af tog på de forskellige depoter (Klampenborg-København-Ballerup)

OM1: Løsningen i forhold til objektfunktionskriteriet for model 1.

OM2: Løsningen i forhold til objektfunktionskriteriet for model 2.

SI: Sidste indsættelsestidspunkt (i forhold til de 8 tidsintervaller).

SK: Løsningen er sekventiel (togene kommer fortløbende efter indsættelse).

2V: Der sendes ca. lige mange tog i begge retninger fra gennemkørende depoter.

B. GAMS-filer

GAMS-kode til løsning af STSP

```

1  SETS
2      i onetrees/ 1,2,3,4,5/
3      j knuder/ 1,2,3,4/ ;
4
5  TABLE
6      a(j,i) distance
7          1      2      3      4
8          1      2      2      2      2
9          2      2      1      2      3
10         3      2      3      3      1
11         4      2      2      1      2 ;
12
13  PARAMETERS
14      c(i)      cost of 1-tree i
15          / 1      22
16           2      23
17           3      24
18           4      16 / ;
19
20  VARIABLES
21      x(i) use 1-tree
22
23      z total distance;
24
25  POSITIVE VARIABLE x ;
26  EQUATIONS
27      objective      the objective function
28      cons1          constraints1 ;
29
30  objective ..      Z=E=sum(i,c(i)*x(i));
31  cons1(j) ..      sum(i,a(j,i)*x(i)) =E= 2;
32
33  MODEL TRANSPORT /ALL/ ;
34  SOLVE TRANSPORT USING LP MINIMIZING Z ;

```

Model 1

```

1  $INCLUDE "C.txt";
2
3  INTEGER VARIABLES
4      DIK(KI)
5      start(k)
6      slut(k)
7  ;
8
9  BINARY VARIABLES
10     x(i,t,k) "Beslutningsvariable (om tog i indsaettes til tiden t paa retningsdepot k).";
11 ;
12
13 VARIABLES
14     tognummer(i,t,k) "Indeholder tognumre for hvert tog til hver tidsenhed paa hvert depot"
15     maxtognummer      "Modellens stoerste tognummer, som objektfunktionen forsoeger at minimere"
16     z                  "Vaerdien af objektfunktionen"
17 ;
18
19 EQUATIONS
20     cons7(i)           "Hvert tog tildeles en togstreng"
21     cons8(t,k)         "Max et tog i hver tidsenhed fra hver station"
22     cons9(KE)          "Indsaet alle udtagne tog igen fra endedepoterne"
23     cons10             "Indsaet alle udtagne tog igen fra gennemkoerende depoter"
24     cons11(KI)         "Sikrer 2V sammen med cons16 og cons17"
25     cons12(KI)         "Sikrer 2V sammen med cons15 og cons17"
26     cons13(KI)         "Sikrer 2V sammen med cons15 og cons16"
27     cons14(i,t,k)      "Sikrer, at tog i fysisk kan indsaettes fra depot k i tidsenhed t"
28     cons15(k)          "Indsaet togene sekventielt fra depot k"
29     cons16(k)          "Indsaet tidligst et tog fra depot k, naar der er en lokofoerer til stede"
30     cons17(i,t,k)      "Fastsaettelse af oevre graense for start(k)"
31     cons18(i,t,k)      "Fastsaettelse af nedre graense for slut(k)"
32     cons19(i,t,k)      "Fastsaettelse af tognumre"
33     cons20(i,t,k)      "Nedre graense for maxtognummer modulo 100"
34     objective          "Objektfunktionen (minimer maxtognummer)"
35 ;
36
37     cons7(i)..         sum((t,k),x(i,t,k)) =E= 1;
38     cons8(t,k)..       sum(i,x(i,t,k)) =L= 1;
39     cons9(KE)..        sum((i,t),x(i,t,KE)) =E= DA(KE);
40     cons10..           sum((i,t,KI),x(i,t,KI)) =E= sum(KI,DA(KI));
41     cons11(KI)..       sum((i,t),x(i,t,KI)) =E= DIK(KI);
42     cons12(KI)..       DIK(KI) =G= FLOOR(DAI(KI)/2);
43     cons13(KI)..       DIK(KI) =L= CEIL(DAI(KI)/2);
44     cons14(i,t,k)..    x(i,t,k) =L= in(i,t,k);
45     cons15(k)..        start(k) + sum((i,t),x(i,t,k))-1 =E= slut(k);
46     cons16(k)..        start(k) =G= klok(k) + 1;
47     cons17(i,t,k)..    start(k) =L= ORD(t) + M * (1 - x(i,t,k));
48     cons18(i,t,k)..    slut(k) =G= ORD(t) - M * (1 - x(i,t,k));
49     cons19(i,t,k)..    tognummer(i,t,k) =E= (initialtog + ctog(k) + ORD(t)) * x(i,t,k);
50     cons20(i,t,k)..    maxtognummer =G= tognummer(i,t,k)-rest(k);
51     objective..        z =E= maxtognummer;
52
53 OPTION reslim = 1000000;
54 OPTION iterlim = 1000000;
55 OPTION optcr = 0.0;
56
57 MODEL reinsection /ALL/;
58
59 SOLVE reinsection USING MIP MINIMIZING z;

```

Data for linie C til Model 1 (C.txt)

```

1  SETS
2      i tog /1,2,3,4,5,6/
3      D depoter /BA,KH,KL/
4      k retningsdepoter /BA,KHS,KHN,KL/
5      KE(k) endedepoter /BA,KL/
6      KI(k) gennemkoerende depoter /KHS,KHN/
7      t mulige tidsenheder til indsaettelse /1*6/
8  ;
9
10 SCALARS
11     M                "Valgt tilpas stor til begraensningerne 17-18" / 10 /
12     initialtog       "Foerste tog der transporterer lokomotivfoerere" / 45125 /
13 ;
14
15 PARAMETERS
16     DA(k)            "Antal tog der skal indsaettes fra depot D"
17                     / BA  3
18                     KHS  1
19                     KHN  1
20                     KL   1 /
21
22     DAI(KI)          "Antal tog der KAN indsaettes fra gennemkoerende depoter"
23                     / KHS  2
24                     KHN  2 /
25
26     in(i,t,k)        "Hvilke tog der er tilgaengelige paa hvilke depoter"
27
28     clok(k)          "Forsinkelse inden lokomotivfoerer kan ankomme til depot k"
29                     / KL   2
30                     KHN  0
31                     KHS  0
32                     BA   2 /
33
34     ctog(k)          "Konstanter der bruges til justering af tognumre paa de enkelte depoter"
35                     / KL  -4899
36                     KHN -5000
37                     KHS -4900
38                     BA  -4999 /
39
40     rest(k)          "Workaround for modulo 100 i begraensning 20"
41                     / KL  40200
42                     KHN 40100
43                     KHS 40200
44                     BA  40100 /
45 ;
46
47 in("1","1","KHN") = 1;
48 in("1","3","KL") = 1;
49 in("1","4","KHS") = 1;
50 in("1","6","BA") = 1;
51
52 in("2","2","KL") = 1;
53 in("2","3","KHS") = 1;
54 in("2","5","BA") = 1;
55 in("2","6","KHN") = 1;
56
57 in("3","1","KL") = 1;
58 in("3","2","KHS") = 1;
59 in("3","4","BA") = 1;
60 in("3","5","KHN") = 1;
61
62 in("4","1","KHS") = 1;
63 in("4","3","BA") = 1;
64 in("4","4","KHN") = 1;
65 in("4","6","KL") = 1;

```

```
66
67   in("5","2","BA") = 1;
68   in("5","3","KHN") = 1;
69   in("5","5","KL") = 1;
70   in("5","6","KHS") = 1;
71
72   in("6","1","BA") = 1;
73   in("6","2","KHN") = 1;
74   in("6","4","KL") = 1;
75   in("6","5","KHS") = 1;
```

Model 2

```

1  $EOLCOM %
2
3  $INCLUDE "C.txt";
4
5  VARIABLES
6      z          "Vaerdien af objektfunktionen." ;
7
8  BINARY VARIABLES
9      x(j)       "Beslutningsvariabel: om vi anvender soejle j eller ej." ;
10
11 EQUATIONS
12     Objective                    "(22): Objektfunktionen."
13     LegalReinsertion(rd,t)       "(23): Togene kommer sekventielt i koereplanen."
14     MaxOneTrainThroughDepot(rd,t) "(24): Hoejst eet tog fra samme depot per tidsenhed og retning."
15     ReinsertedTrains(d)          "(25): Samme antal indsaettes, som der blev udtaget per depot.";
16
17     Objective..                  z =E= sum(j,c(j)*x(j));
18     LegalReinsertion(rd,t)$ (ord(t)<card(t)).. sum(j,a(rd,t,j)*x(j)) =L= sum(j,a(rd,t+1,j)*x(j));
19     MaxOneTrainThroughDepot(rd,t).. sum(j,a(rd,t,j)*x(j)) =L= 1;
20     ReinsertedTrains(d)..        sum(j,assignDepot(d,j)*x(j)) =E= insertIn(d);
21
22 MODEL reinsertion / ALL /;
23
24 OPTION reslim = 1000000;
25 OPTION iterlim = 1000000;
26 OPTION optcr = 0.0;
27
28 SCALAR S, n1, n2, n3, p, q, temp, SolutionNo;
29 S = 6;
30
31 FILE reins /"reinsC2009_v1.prn"/;
32 reins.pw = 300;
33 reins.ps = 3000;
34 reins.nd = 0;
35 PUT reins;
36
37 FOR (p = 0 TO S BY 1,
38     n1 = p;
39     FOR (q = 0 TO (S-p) BY 1,
40         n2=q;
41         n3 = S-p-q;
42         insertIn("DepotBA")=n1;
43         insertIn("DepotKH")=n2;
44         insertIn("DepotKL")=n3;
45
46         SOLVE reinsertion MINIMIZING z USING MIP;
47
48         temp = 0;
49         SolutionNo = 100*insertIn("DepotBA")+10*insertIn("DepotKH")+insertIn("DepotKL");
50         PUT "Solution:      ",SolutionNo/;
51         PUT "Objective value:", z.l/;
52         temp = MOD(z.l,100);
53         PUT "OPT-M2:      ", temp/;
54         LOOP((j,rd,d),
55             IF (x.l(j)>0 AND assignDepot(d,j)=1 AND assignDirection(rd,j)=1,
56                 PUT @1 rd.tl;
57                 PUT @7 stamme(j);
58                 PUT @27 c(j)/;
59             );
60         );
61         PUT "-----"/;
62     );
63 );
64
65 PUTCLOSE reins;

```

Data for linie C til Model 2 (C.txt)

```

1 SETS
2     depotReinsertionDirections /BA, KHS, KHN, KL/           % Retningsdepoter
3     depots /DepotBA, DepotKH, DepotKL/                     % Depoter
4     timeSlots /t1*t6/                                       % Tidsenheder
5     columns /j1*j24/                                         % Soejler
6 ;
7
8 ALIAS
9     (depotReinsertionDirections, rd, rdd)                   % Alias for retningsdepoter
10    (depots, d, dd)                                          % Alias for depoter
11    (timeSlots, t, tt)                                       % Alias for tidsenheder
12    (columns, j, jj)                                         % Alias for soejler
13 ;
14
15 PARAMETERS
16     insertIn(d) / DepotBA 3, DepotKH 2, DepotKL 1 /       % Svarer til d_k i modellen
17     c(j)                                                  % Samme som i modellen
18     assignDirection(rd,j)                                  % Reelt overfloedig - bruges kun i output
19     assignDepot(d,j)                                       % Svarer til b-matricen i modellen
20     a(rd,t,j)                                              % Samme som i modellen
21     stamme(j)                                              % Reelt overfloedig - bruges kun i output
22 ;
23
24 stamme("j1")=3;
25 stamme("j2")=4;
26 stamme("j3")=5;
27 stamme("j4")=6;
28 stamme("j5")=1;
29 stamme("j6")=2;
30 stamme("j7")=6;
31 stamme("j8")=1;
32 stamme("j9")=2;
33 stamme("j10")=3;
34 stamme("j11")=4;
35 stamme("j12")=5;
36 stamme("j13")=6;
37 stamme("j14")=1;
38 stamme("j15")=2;
39 stamme("j16")=3;
40 stamme("j17")=4;
41 stamme("j18")=5;
42 stamme("j19")=3;
43 stamme("j20")=4;
44 stamme("j21")=5;
45 stamme("j22")=6;
46 stamme("j23")=1;
47 stamme("j24")=2;
48
49 c("j1")=3;
50 c("j2")=4;
51 c("j3")=5;
52 c("j4")=6;
53 c("j5")=7;
54 c("j6")=8;
55 c("j7")=1;
56 c("j8")=2;
57 c("j9")=3;
58 c("j10")=4;
59 c("j11")=5;
60 c("j12")=6;
61 c("j13")=3;
62 c("j14")=4;
63 c("j15")=5;
64 c("j16")=6;
65 c("j17")=7;

```

```

66 c("j18")=8;
67 c("j19")=1;
68 c("j20")=2;
69 c("j21")=3;
70 c("j22")=4;
71 c("j23")=5;
72 c("j24")=6;
73
74 a("BA", "t1", "j1")=1;
75 a("KHN", "t4", "j1")=1;
76 a("KL", "t4", "j1")=1;
77 a("BA", "t2", "j2")=1;
78 a("KHN", "t5", "j2")=1;
79 a("KL", "t5", "j2")=1;
80 a("BA", "t3", "j3")=1;
81 a("KHN", "t6", "j3")=1;
82 a("KL", "t6", "j3")=1;
83 a("BA", "t4", "j4")=1;
84 a("BA", "t5", "j5")=1;
85 a("BA", "t6", "j6")=1;
86 a("KHN", "t1", "j7")=1;
87 a("KL", "t1", "j7")=1;
88 a("KHS", "t4", "j7")=1;
89 a("BA", "t4", "j7")=1;
90 a("KHN", "t2", "j8")=1;
91 a("KL", "t2", "j8")=1;
92 a("KHS", "t5", "j8")=1;
93 a("BA", "t5", "j8")=1;
94 a("KHN", "t3", "j9")=1;
95 a("KL", "t3", "j9")=1;
96 a("KHS", "t6", "j9")=1;
97 a("BA", "t6", "j9")=1;
98 a("KHN", "t4", "j10")=1;
99 a("KL", "t4", "j10")=1;
100 a("KHN", "t5", "j11")=1;
101 a("KL", "t5", "j11")=1;
102 a("KHN", "t6", "j12")=1;
103 a("KL", "t6", "j12")=1;
104 a("KL", "t1", "j13")=1;
105 a("KHS", "t4", "j13")=1;
106 a("BA", "t4", "j13")=1;
107 a("KL", "t2", "j14")=1;
108 a("KHS", "t5", "j14")=1;
109 a("BA", "t5", "j14")=1;
110 a("KL", "t3", "j15")=1;
111 a("KHS", "t6", "j15")=1;
112 a("BA", "t6", "j15")=1;
113 a("KL", "t4", "j16")=1;
114 a("KL", "t5", "j17")=1;
115 a("KL", "t6", "j18")=1;
116 a("KHS", "t1", "j19")=1;
117 a("BA", "t1", "j19")=1;
118 a("KHN", "t4", "j19")=1;
119 a("KL", "t4", "j19")=1;
120 a("KHS", "t2", "j20")=1;
121 a("BA", "t2", "j20")=1;
122 a("KHN", "t5", "j20")=1;
123 a("KL", "t5", "j20")=1;
124 a("KHS", "t3", "j21")=1;
125 a("BA", "t3", "j21")=1;
126 a("KHN", "t6", "j21")=1;
127 a("KL", "t6", "j21")=1;
128 a("KHS", "t4", "j22")=1;
129 a("BA", "t4", "j22")=1;
130 a("KHS", "t5", "j23")=1;
131 a("BA", "t5", "j23")=1;
132 a("KHS", "t6", "j24")=1;

```

```

133 a("BA","t6","j24")=1;
134
135 assignDirection("BA","j1")=1;
136 assignDirection("BA","j2")=1;
137 assignDirection("BA","j3")=1;
138 assignDirection("BA","j4")=1;
139 assignDirection("BA","j5")=1;
140 assignDirection("BA","j6")=1;
141 assignDirection("KHN","j7")=1;
142 assignDirection("KHN","j8")=1;
143 assignDirection("KHN","j9")=1;
144 assignDirection("KHN","j10")=1;
145 assignDirection("KHN","j11")=1;
146 assignDirection("KHN","j12")=1;
147 assignDirection("KL","j13")=1;
148 assignDirection("KL","j14")=1;
149 assignDirection("KL","j15")=1;
150 assignDirection("KL","j16")=1;
151 assignDirection("KL","j17")=1;
152 assignDirection("KL","j18")=1;
153 assignDirection("KHS","j19")=1;
154 assignDirection("KHS","j20")=1;
155 assignDirection("KHS","j21")=1;
156 assignDirection("KHS","j22")=1;
157 assignDirection("KHS","j23")=1;
158 assignDirection("KHS","j24")=1;
159 assignDepot("DepotBA","j1")=1;
160 assignDepot("DepotBA","j2")=1;
161 assignDepot("DepotBA","j3")=1;
162 assignDepot("DepotBA","j4")=1;
163 assignDepot("DepotBA","j5")=1;
164 assignDepot("DepotBA","j6")=1;
165 assignDepot("DepotKH","j7")=1;
166 assignDepot("DepotKH","j8")=1;
167 assignDepot("DepotKH","j9")=1;
168 assignDepot("DepotKH","j10")=1;
169 assignDepot("DepotKH","j11")=1;
170 assignDepot("DepotKH","j12")=1;
171 assignDepot("DepotKL","j13")=1;
172 assignDepot("DepotKL","j14")=1;
173 assignDepot("DepotKL","j15")=1;
174 assignDepot("DepotKL","j16")=1;
175 assignDepot("DepotKL","j17")=1;
176 assignDepot("DepotKL","j18")=1;
177 assignDepot("DepotKH","j19")=1;
178 assignDepot("DepotKH","j20")=1;
179 assignDepot("DepotKH","j21")=1;
180 assignDepot("DepotKH","j22")=1;
181 assignDepot("DepotKH","j23")=1;
182 assignDepot("DepotKH","j24")=1;

```

C. Godkendt projektbeskrivelse

Projektbeskrivelse for virksomhedsprojekt

Projektdeltagere:

Christian Mikkelsen Jensen (XXXXXX-XXXX)

Anders Bjerg Pedersen (XXXXXX-XXXX)

Dansk titel: "Analyse af genindsættelse af tog i DSB S-tog"

Engelsk titel: "Analysis of reinsertion of trains at DSB S-tog"

Varighed: Blok 4 og blok 1 2009 (halv tid over to blokke).

Intern vejleder: Pawel Winter (DIKU)

Ekstern vejleder: Jens Clausen (DSB S-tog)

Afleveringsfrist: Fredag den 30. oktober 2009

Projektbeskrivelse:

Projektet tager udgangspunkt i et konkret dagligdagsproblem hos DSBs S-togsafdeling. Når der opstår en form for forstyrrelse (enten teknisk, mandskabsmæssig eller andet), tages ofte en hel toglinje ud af gangen. Problemet går ud på at finde den optimale måde at genindsætte de udtagne tog hurtigst muligt. Dette problem har DSB S-tog fundet to mulige løsninger på i form af matematiske modeller, der hver især har deres styrker og svagheder. Overordnet går opgaven ud på at sammenligne, teste og eventuelt udvikle de to metoder med henblik på forbedringer og større anvendelighed.

Projektets faglige delelementer bygger på operationsanalyse, og det planlagte indhold er som følger:

- Grundig beskrivelse af DSB S-tog: organisation, arbejdsopgaver, rutiner, arbejdsmiljø og lignende; med specielt fokus på Analyseafdelingen, hvor vores problemstillinger bliver bearbejdet.
- Teoriafsnit omkring søjlegenerering og dekomposition, herunder Danzig-Wolfe-dekomposition.
- Beskrivelse af det aktuelle problem og grundene til, at det er væsentligt og nødvendigt at anvende matematisk programmering til løsningen.
- Beskrivelse og sammenligning af de to løsningsmodeller; parametre, objektfunktioner, begrænsninger med mere.
- Mulige udvidelsesmuligheder eller eventuelt samkøring af modellerne.
- Implementering, afprøvning og test af udgangsmodellerne samt eventuelle videreudviklede modeller.