

SICS Technical Report
T2000:07

ISRN : SICS-T--2000/07-SE
ISSN : 1100-3154

ACoor Rapport 2 **Översikt av tekniker och metoder**

by

Per Kreuger, Martin Aronsson, Per Holmberg and Simon Lindblom

March 2000

Swedish Institute of Computer Science
Box 1263, SE-164 29 Kista, SWEDEN

ACOOOR Rapport 2

Översikt av tekniker och metoder

Per Kreuger, Martin Aronsson, Per Holmberg, Simon Lindblom

2000-03-20
SICS Technical Report T2000/07

Sammanfattning

I arbetet med att modellera ett produktionsplaneringsproblem i järnvägsindustrin har vi identifierat ett antal metoder och tekniker som vi hävdar är användbara även utanför järnvägsdomänen. I denna rapport presenterar vi kort problemställningarna i järnvägsdomänen, presenterar de viktigaste av de tekniker vi identifierat, använt och utvecklat samt indikerar en del andra problemområden för vilka vi bedömer att dessa tekniker skulle kunna användas.

Nyckelord Schemaläggning, omlopp för fordon och personal, tekniker för planering av tågrörelser

ISSN 1100-3154
ISRN: SICS-T-2000/07-SE

1 Inledning och bakgrund

I ACOOR-projektet används en rad tekniker och metoder för att lösa en konkret uppsättning problem kring planering av effektiv tjänsteproduktion för spårbundna transporter. Problemet präglas såväl av ett mer eller mindre fixerat spårnät som av den uppdelning i delproblem som man traditionellt arbetar med inom produktionsplanering inom järnvägsindustrin.

Detta dokument sammanställer och beskriver de tekniker som används inom projektet samt pekar på andra problemområden där denna typ av tekniker också kan komma till användning. Dokumentet är ett projektresultat för projektet ACOOR under 1999/2000.

2 Problemställning

I ACOOR-projektet angriper vi problemet att samordna planeringsprocesserna för ett antal resurser som krävs för att utföra transporttjänster i ett järnvägsnät. Vi avser därvid att och minska de totala kostnaderna för användningen av dessa resurser.

De tre viktigaste av dessa resurser är spårlägen, fordon samt resande personal.

2.1 Produktionsplanering inom järnvägsindustrin

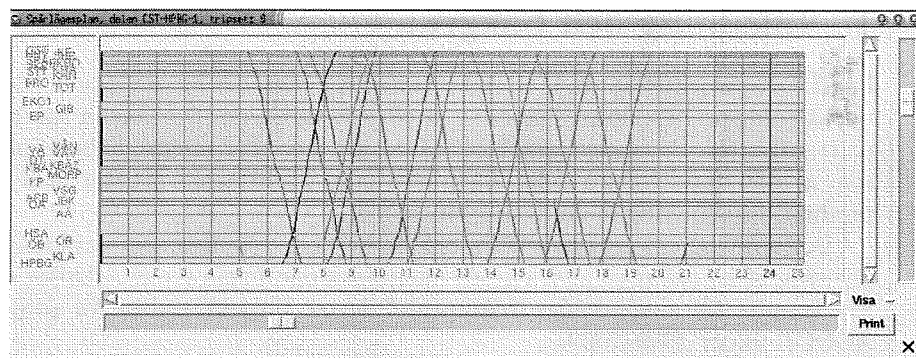
Produktionsplanering inom järnvägsindustrin kan sägas bestå i att planera tågrörelser givet en specifikation av (tåg-) *tripp*. Varje tripp måste för att kunna utföras allokeras spårutrymme under vissa tidsintervall (spårlägen), fordonsresurser som kan utföra rörelsen samt resande personal, främst lokförare och konduktörer.

Vi antar att de trippar som behövs för att tillfredsställa ett visst transportbehov i förväg bestämts och representeras t.ex. som ett antal trippar med krav på avgångs- och ankomsttider. Vi kallar denna parameter en trippspecifikation. Vi antar vidare att den väg som skall traverseras av varje given tripp är fullständigt bestämd av specifikationen. Idag löses problemet att generera en sådan specifikation genom att vi tillhandahåller ett språk i vilket man kan uttrycka cykliska förlopp. Specifikationerna har hittills generats för hand vilket är relativt arbetskrävande. Vi utvecklar därför också ett grafiskt verktyg (SpecGUI) som kan stödja denna process.

I allmänhet skulle man vilja generera specifikationer utgående från beskrivningar av transportbehov givna som flöden. Vi har dock idag inte ambitionen att gå vidare med detta problem. Metoder för att lösa sådana problem har utvecklats inom operationsanalysen. För en utmärkt översikt se [BWZ97].

2.2 Delproblem

Man kan se de ovan beskrivna problemen som tre separata allokeringsproblem. Dessa kan lösas vart och ett för sig men flera egenskaper hos lösningarna är



Figur 1: Spårlägesplan

korrelerade. Traditionellt löser man de tre delproblemen i sekvens i en *vattenfallsmodell* med viss manuell utförd återkoppling.

Först allokeras till varje tripp ett antal spårågen, vilket kan modelleras som ett mer eller mindre klassiskt schemalägningsproblem [Gos93], [CP89], [AC91] med krav på exklusivt utnyttjande av spårresurserna. Resultatet av denna fas av planeringsarbetet är traditionellt en fullständigt bestämd tidtabell.

Därpå löses problemet att fastställa *lokomlopp* som beskriver lokens väg genom nätet och tillser att varje tripp är försörjd med lok från avgång till ankomst. Detta innefattar också att fastställa *passiva transporter* av lok där ett lok flyttas från ankomstplatsen för en tripp till avgångsplatsen för nästa i dess omlopp. Denna *passiva* transport kan antingen ske genom att loket själv förflyttar sig via lediga spårågen till utgångsplatsen för sitt nästa uppdrag (s.k. "deadhead") eller genom att loket under vissa förutsättningar medföljer en redan fastlagd tripp ("twohead"). Det senare är oftast att föredra då kostnaderna för en sådan passiv transport oftast är lägre än för "deadheads".

Detta problem kan lösas med traditionella optimeringstekniker givet att man dessförinnan genererat en fullt bestämd tidtabell. Om inte blir problemet dessvärre avsevärt mycket svårare.

Till sist behöver man också generera *personalomlopp*. På samma sätt som i fallet med loken måste personalen finnas på plats vid rätt tid för att trippen skall kunna genomföras. Även personal måste ibland resa "passivt" t.ex. hem vid dygnsvila. Utöver de krav som ställs på lokomloppen tillkommer begränsningar på de slingor (tjänstgöringsperioder) som utgör delar av omloppen. Dessa är t.ex. begränsade i längd (typiskt på ett till två dygn) vilket komplicerar problemet avsevärt. Vidare finns en rad ytterligare begränsningar på personalomloppen. Dessa är beroende på lagstiftning, avtal och lokalt varierande praxis.

Sammantaget gör dessa begränsningar personalplaneringsproblemet till det mest komplexa av delproblemen i produktionsplanering för järnvägsindustrin.

abstraktion som är ett av projektets huvudteman.

Behovet av att samplanera användningen av flera separata resurser för varje uppdrag gör att samordningsproblematiken hamnar i fokus. Dels uppstår p.g.a. problemets storlek ett behov av att samordna delplaner för vart och ett av de tre resursproblemen, dels finns det stora vinster att göra på att bryta upp den strikta ordning mellan planeringsstegen som idag är den dominerande ansatsen för att lösa det totala problemet.

Ett exempel på detta förhållande är följande:

Antag att en planspecifikation innehåller ett visst spel ("slack") i tidsangivelserna för avgångar och ankomster.

Antag vidare att man fixerar en lösning på spårlägesproblemet som innebär att vissa avgångar fastställs till tidpunkter som knappt föregår vissa ankomster till avgångsplatserna för de förra.

Detta innebär att samma fordon inte kan användas för att betjäna vissa sekvenser av fordonsuppdrag vilket i sin tur kan spolia möjligheten att effektivt utnyttja fordonsresurserna.

Att bibehålla så mycket som möjligt av spelet i avgångstiderna ger uppenbarligen i det allmänna fallet utrymme för att generera "bättre" omlopp. Tyvärr bryter de mycket effektiva traditionella metoder (nätverksflöden) för att generera omlopp samman då tiderna tillåts variera. Detta gäller i allmänhet då frågan om huruvida ett omlopps-uppdrag (tripp) kan följas av ett annat i en omloppsarbete (slinga) inte kan avgöras lokalt, d.v.s. med kunskap om bara dessa två uppdrag.

Det är därför nödvändigt att använda heuristiska metoder för att generera omlopp som kan användas för att inskränka (delvis ordna) specifikationen så att de spårlägesplaner som genereras i enlighet med denna inskränkta specifikation tillåter oss att bilda de slingor som i någon mening är lokalt "goda". För dessa spårlägesplaner kan man sedan generera optimala omlopp på traditionellt sätt.

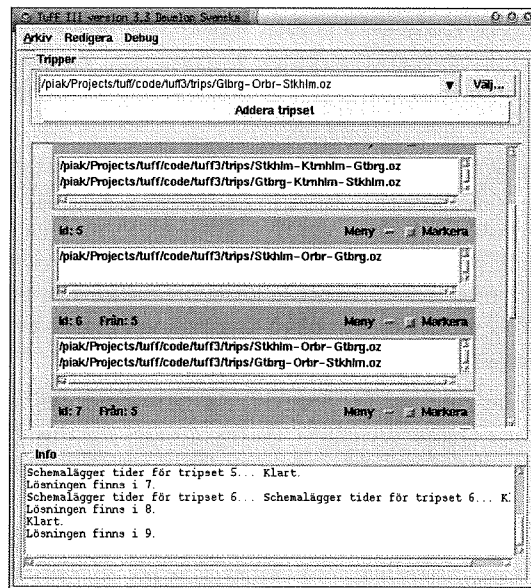
Denna typ av problemlösningstrategi där man delvis löser ett delproblem för att inskränka planeringsutrymmet för ett senare steg i processen kallas ibland iterativ problemlösning.

Vi bygger f.n. ett interaktivt verktyg som innehåller specialiserade delproblemlösare samt mekanismer för att dela upp, abstrahera och förfina problemspecifikationer. I ett allmänt perspektiv kan de grundläggande mekanismer vi därvid implementerar utgöra kärnan i ett ramverk för beslutstödssystem för en stor klass av komplexa tekniska system.

3 Grundtekniker

3.1 Villkorsprogrammering ("constraint programming")

Villkorsprogrammering (CP) [vH89], [Tsa93], [Sim96] bygger på idén om en abstrakt rymd av påståenden, inskränkningar eller villkor (constraint space).



Figur 3: Grafiskt gränssnitt till en samordningskomponent

Vissa av dessa påståenden kan anses vara fullt bestämda. Tag t.ex. påståendet att ett givet tåg skall avgå Avesta 15.05 torsdag 15/4 år 2001. Andra påståenden är mindre exakta t.ex. det att Ovako behöver transportera mellan 320 och 380 kiloton stål från Hofors och Hellefors till Malmö nästa år. Båda dessa påståenden kan representeras som *villkor* ett villkorprogrammeringsystem. Vi kallar dem villkor eller inskränkningar på värderymden för de variabler som påståendet innehåller.

Det är naturligtvis icke-trivialt att i en rymd av sådana påståenden avgöra hur t.ex. dessa båda påståenden är relaterade givet någon matematisk modell av ett planeringsproblem men under vissa omständigheter går det att räkna på sådana abstrakta objekt och avgöra t.ex. motsägelsefrihet, d.v.s. att avgöra om påståendena är förenliga.

Det går också att beräkna ett eller flera *vittnen*, d.v.s. tilldelningar av värden till alla variabler i en rymd. Detta kallas att räkna upp ("enumerate") *sökrymden* för ett visst problem och innefattar i allmänhet sökning. Sådana vittnen kan i produktionsplaneringsdomänen t.ex. utgöras av konkreta produktionsplaner. Om flera sådana tas fram kan de jämföras med avseende på olika kostnadsått. Att hitta den i någon mening bästa planen modelleras som ett optimeringsproblem i villkorssystemet.

Mycket av sökningen i en uppräknings av sökrymden för ett givet problem kan i allmänhet elimineras med en teknik som kallas villkorspropagering. Detta innebär i korthet att varje påstående som inte är helt bestämt betraktas som en tillfälligt avbruten beräkning som kan fås att interagera med andra samtida sådana beräkningar. Beräkningar avbryts när den information som behövs för

att bestämma ett värde saknas men återupptas om den senare blir tillgänglig. Man kan se villkorsprogrammeringssystemet som en mängd parallella processer som kommunicerar, interagerar och synkroniseras via delade variabler i en dataflödesgraf.

Huruvida det är möjligt att beräkna lösningar på ett givet problem eller inte bygger mycket på uttryckskraften i det språk som används för att uttrycka påståenden om problemet. Att i ett sådant språk formulera en matematisk modell av någon verklig process är i det allmänna fallet ett mycket svårt problem. Trots detta har man med dessa tekniker lyckats mycket bra med att modellera och lösa erkänt svåra planeringsproblem t.ex. många klassiska schemalägnings- och resursallokeringsproblem.

Tag exemplet med en avgångstid. Är den känd skulle vi vilja representera den som ett tal i någon tidsenhet. Antag nu att den inte är känd, men det ändå finns viss information om den, t.ex. att den måste ligga mellan 11 och 12 någon given dag och att den måste följa efter ankomsttiden för någon annan tripp.

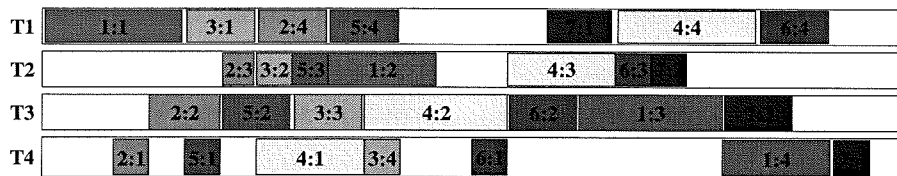
Detta är en mycket stark inskränkning av värdet på avgångstiden. Ändå innehåller den en osäkerhet som skiljer den på ett fundamentalt sätt från den helt bestämda. Ett antal sådana påståenden kan t.ex. uttrycka relationen mellan denna avgångstid och andra avgångstider eller ankomster och kan sägas representera en specifikation av en produktionsplan. Ju mer bestämd den är desto närmare ligger den en färdig plan med helt fastställda tider.

Man kan i princip konstruera en produktionsplan genom att successivt addera mer och mer information till en villkorsrymd.

3.1.1 Ändliga domäner (“finite domains”)

De villkorsprogrammeringssystem som mest aktivt utvecklats de senaste 10 åren är de som bygger på *ändliga domäner*. I ett sådant system kan varje variabel anta värden ur en ändlig mängd diskreta värden. Denna typ av variabler är naturliga att använda för att modellera diskreta storheter som antal fordon eller personer som tilldelats ett visst uppdrag. De är däremot onödigt restriktiva då man modellerar storheter som kan antas variera över kontinuerliga domäner (med oändligt antal möjliga värden), t.ex. tid.

Tekniker från operationsanalysen, t.ex. linjärprogrammering (LP) och heltalsprogrammering (IP) hanterar effektivt modeller där de flesta variablerna är kontinuerliga och där en enkel och väldefinierad kostnadsfunktion väl fångar “godheten” hos skilda lösningar av ett givet problem. De tekniker som tagits fram inom villkorsprogrammering med ändliga domäner fungerar däremot väl även då en majoritet av variablerna modellerar naturligt diskreta storheter, då kostnadsfunktionen är svärfångad samt när modellen innehåller komplexa (t.ex. ickelinjära) villkor. På så sätt kan dessa två klasser av tekniker sägas komplettera varandra.



Figur 4: Gantt-schema för 6 arbeten bestående av 24 uppdrag fördelade på 4 resurser

3.1.2 Globala villkor

Den första typ av villkor som studerades inom villkorsprogrammeringen var sådana som relaterade två enskilda variabler, t.ex. $<$, \leq , \neq , $=$ etc. I kontrast till dessa enkla binära villkor har man på senare år fokuserat mer och mer på mer komplexa villkor mellan ett obegränsat antal variabler. Exempel på sådana villkor är sådana som relaterar variabler med värdet hos en linjär summa eller upprätthåller parvis olikhet hos en godtycklig mängd variabler.

Sådana villkor kan i princip oftast kodas i termer av en mängd enklare binära villkor som semantiskt har samma mening. Detta är dock sällan praktiskt då en effektiv lösning många gånger är beräkningsmässigt alltför komplex för att kunna realiseras genom att endast beakta variablerna parvis. Uttrycket "globala villkor" för denna typ av villkor introducerades i [BC94] och refererar till de resonemang som kan föras över ett flertal variabler som relateras i ett ickebinärt villkor.

Globala villkor utgör abstraktioner av mera komplexa egenskaper hos problem och möjliggör beräkningar på en mer detaljerad modell. Många gånger kan metoder från operationsanalysen eller teorin för algoritmer som verkar på grafer effektivt och naturligt integreras i ett villkorsprogrammeringssystem som globala villkor. Detta är ett aktivt och mycket lovande forskningsområde inom villkorsprogrammering. För en systematiskt beskrivning av ett stort antal globala villkor se [Bel00].

3.1.3 Schemaläggning

Till de exempel på globala resonemang som framgångsrikt introducerats i villkorsprogrammeringssystem hör en rad grundläggande schemaläggningmekanismer.

Ett schemaläggningssystem består av ett antal *uppdrag* ("tasks") med begränsningar på *starttider*, *sluttider* och *tidsutsträckning*. Ofta är uppdragen *partiellt ordnade* i sekvenser. En totalt ordnad delmängd av uppdrag kallas ofta för ett *arbete* ("job"). Varje uppdrag *belägger* eller utnyttjar en eller flera resurser under vissa tidsintervall.

Resurser kan i allmänhet modellera vitt skilda företeelser.

T.ex:

- bearbetningsutrustning i en framställningsprocess
- personal eller fordon i ett godstransportnät
- en router i ett kommunikationsnätverk

Inom järnvägsindustrin är de viktigaste resurserna spår, fordon och personal.

Att ordna uppdragen så att inga resursbegränsningar överskrids kallas att *schemalägga* uppdragen och är i det allmänna fallet ett mycket svårt beräkningstekniskt problem. Icke desto mindre gör de många praktiska tillämpningarna för metoder inom detta område det relativt väl studerat.

Flera av de bästa ansatserna för att lösa schemalägningsproblem under de senaste tio åren har introducerats som globala villkor i villkorsprogrammerings-system [CP89], [CP94], [MS94], [BLP95], [CL96].

3.1.4 Omlopp

Vissa typer av arbeten kan uppfattas som cykliska förlopp eller flöden. I ett enkelt fordonsproblem kan fordonens rörelser uppfattas som cykliska vägar i en graf. Hur svårt det är att lösa sådana problem beror starkt på de villkor man ställer på bågarna i grafen.

Om villkoret kan avgöras lokalt kan problemet modelleras som ett *nätverksflöde* [AMO93]. För sådana problem finns mycket effektiva optimerande lösare (se nedan avsnitt 3.2.1) Sådana tekniker används praktiskt i framtagningen av fordonsomlopp [DHKK97].

Om en båge i grafen representerar en *vändning* av ett lok från en tripp till en annan är modellen ett klassiskt nätverksflöde endast om avgörandet om bågen tillfredsställer modellen kan göras lokalt, dvs. oberoende av andra uppdrag (tripp) än de två som bågen binder samman. Detta är i praktiken sällan fallet. Inom produktionsplanering för järnvägsindustrin förekommer två viktiga omloppsproblem som inte har denna egenskap.

- Problemet att generera lokomlopp utan ett temporalt fixerat schema
- Problemet att generera cykler av bestämd "längd" t.ex. arbetspass med dygnsvila

Mycket av arbetet som skett i projektet har gått ut på att karaktärisera dessa problem. Följande referenser sätter problemet i kontext [BG81], [Sav85], [GA86], [Sol86], [Sol87], [SD88], [PR93], [TP93]. Vi tror att mekanismer för att lösa denna typ av problem kan vara av stort värde för beslutsstödssystem i en vitt skilda domäner (se avsnitt 4).

3.2 OR-tekniker

Man bör jämföra villkorsprogrammering med traditionella optimeringstekniker som framkommit inom operationsanalysen. Dessa tekniker bygger oftast på massiva beräkningar där indata ges i form av en fullständig problembeskrivning.

Dessa metoder är mycket effektiva när modellen av det verkliga problemet passar väl in i någon välkänd klass av problem och problemet är renodlat, dvs. oberoende från ett svårbeskrivbart eller alltför komplext kontext. De är därför många gånger mindre lämpliga i ett tidigt stadium av planeringsprocessen då många parametrar ännu är okända.

Däremot har de sin givna plats när problemrymden krympts med andra metoder och när man vill jämföra resultatet av strategiska val i väl avgränsade delproblem. Man kan se många av dessa metoder som planeringsprimitiver, metoder som kan användas för att utforska egenskaper hos det problem man ställs inför i en given situation.

Villkorstekniken, däremot, erbjuder möjligheten att upprätthålla en dynamiskt föränderlig rymd av påståenden som representerar alla för ögonblicket möjliga (del-)planer givet de villkor som kundkrav, resursbegränsningar och kostnadshänsyn sätter. Vi kan välja att optimera delar av planen sent och inkrementellt och lämna resten av planen bara delvis bestämd. En annan fördel är att det är möjligt att inkorporera tekniker från operationsanalysen i villkorsparadigmet i form av globala villkor. Detta gör villkorsprogrammering till ett *”integrativt”* projekt, där tekniker från skilda fält sammanställs och görs tillgängliga för modelleringsexperter utan att dessa avkrävs detaljerad algoritmisk kompetens. Detta ser vi som en av de stora fördelarna med villkorstekniken. Några av de tekniker vi ser omedelbar användning för i produktionsplaneringsdomänen sammanfattas nedan.

3.2.1 Linjärprogrammering

Linjärprogrammering används för att modellera optimeringsproblem där kostnadsfunktionen kan uttryckas som en linjär summa över kontinuerliga variabler och där villkoren är linjära olikheter över dessa.

Exempel: Linjärt program

Maximera

$$c_1x_1 + \dots + c_nx_n$$

där

$$a_{11}x_1 + \dots + a_{1n}x_n \leq b_1$$

⋮

$$a_{m1}x_1 + \dots + a_{mn}x_n \leq b_m$$

och

$$x_1 \geq 0, \dots, x_n \geq 0$$

SIMPLEX SIMPLEX är den äldsta och mest kända algoritmen för att lösa denna typ av problem. Trots att SIMPLEX i de flesta tillämpningar har mycket goda komplexitetsegenskaper (polynomiell tidskomplexitet) finns för vissa klasser av problem specialiserade algoritmer som är ändå effektivare. Ett exempel på sådana algoritmer är *nätverksflödesoptimering* (se nedan).

Den stora nackdelen med SIMPLEX är att den fungerar bara om alla villkoren är linjära och variablerna är kontinuerliga. Det senare kravet utesluter alla disjunktiva villkor (beslutsproblem) där värdet på en funktion beror på en boolsk variabel. Det forskningsfält som studerar de problem som uppstår då man lättar på det första kravet kallas inom OR för ickelinjär programmering och kommer ej närmare att beröras i denna översikt. De problem som kan formuleras då variabler tillåts variera över diskreta värden studeras inom området heltalsprogrammering. Ibland kallas problem där båda typerna av variabler förekommer blandade ("mixed").

Nätverksflöden Nätverksflöden är ett samlingsbegrepp för många typer av linjärprogrammeringsalgoritmer som hanterar optimering av flöden i nätverk. Området är väl utforskat och många problem är klassificerade. Tre viktiga delproblemen i nätverksflödesoptimering är *kortaste väg*, *maximalt flöde* och *minimal kostnad*. I beräkning av kortaste väg gäller det att hitta kortaste vägen mellan några eller alla punkter i ett nätverk. Maximala flöden kodar problem där flödet mellan par av punkter i nätverket ska maximeras. I minimala kostnadsproblem ansätts kostnader på flödet mellan olika punkter och det gäller att minimera kostnaden att skicka flöden från en eller flera punkter i nätverket till en eller flera destinationer.

Eftersom detta område är stort så skiljer sig algoritmerna ganska mycket från varandra. När det gäller minimala kostnadsproblem används ofta specialiserade varianter av SIMPLEX.

3.2.2 Heltalsprogrammering

Många gånger kan man undvika att introducera heltalsvariabler genom att använda mer eller mindre avancerade modelleringstrick som kodar ett problem med heltalskrav som ett linjärt problem. Detta kräver dock ofta en förhållandevis specialiserad matematisk kompetens och är långt ifrån alltid möjligt.

Om man tillåter heltalskrav får man dock en betydligt friare modellering men tvingas istället introducera sökning i lösningsalgoritmerna. De sökmekanismer som utvecklats för att lösa denna typ av problem använder på ett direkt sätt kostnadsfunktionen och löser i varje steg (iteration) *linjära relaxeringar* (delar av) av det verkliga problemet.

Ett stort antal heuristiska metoder har utvecklats för att få sökningen att konvergera snabbare för vissa klasser av problem. Att givet ett konkret problem avgöra om det kan modelleras som en instans av ett av dessa väl studerade problemklasser kräver i allmänhet också specialiserad matematisk kompetens.

Lagrange-relaxering Lagrange-relaxering är en teknik som går ut på att på ett systematiskt sätt omformulera heltalskraven som parametrar i kostnadsfunktionen i ett motsvarande linjärt program. Dessa parametrar justeras därpå genom att det relaxerade problemet löses med avseende på den parametriserade kostnadsfunktionen. I varje steg justeras parametrarna baserat på resultaten för den föregående iterationen.

	LP (linj.progr.)	IP (heltalsprogr.)	CP (villkorsprogr.)
variabeldomäner	kontinuerliga	heltal/blandade	heltal (ändl. mängd), intervall
kostnadsfunktion	optimering	optimering, sökning	domänbegränsning
skalbarhet	polynomiell kompl.	vissa problemklasser	bra för beslutsproblem
sökning	SIMPLEX	problemberoende	spec. metoder integreras

Tabell 1: Några egenskaper hos ett urval tekniker

Lagrangerelaxering har framgångsrikt används för att lösa t.ex. mycket stora fordonssomloppsproblem [Löb98] och är också en av de viktigaste teknikerna för att lösa parningsproblem (se nedan). För en allmän introduktion till Lagrange-relaxering se [Ree95a]. Denna referens [Ree95b] innehåller dessutom beskrivningar av ett antal s.k. lokala sökmekanismer som ej behandlats i denna översikt.

Parningsalgoritmer (“Pairing”) De flesta optimerande system för personalplanering inom transportsektorn (dvs resande personal) som tagits fram arbetar i två steg:

1. Tag fram ett antal möjliga slingor, d.v.s. arbeten bestående av uppdrag (ben) som kan utföras i ordning av t.ex. en person; Varje slinga skall dessutom (ibland genom att lägga till passiva resor) beskriva en cykel i spårnätsgrafen, d.v.s. t.ex. starta och sluta på samma ort
2. Lös ett optimeringsproblem som består i att bland de ovan genererade slingorna välja en delmängd så att alla uppdrag ingår i minst en cykel och så att en global kostnad minimeras

Steg två är ur matematisk synpunkt enkel att formulera, även om problemens storlek många gånger kan göra dem svårlösta. För att nå så goda resultat som möjligt gäller det därför att i steg 1 generera så många kandidater som man beräkningsmässigt kan hantera i steg 2. Det är dock i praktiken oftast omöjligt att beakta alla möjliga slingor, så vilket urval som görs i steg 1 är av avgörande betydelse för resultatet.

Steg 2 kan förenklat beskrivas som att man bygger upp en matris, med uppdragen som rader och de i steg 1 genererade slingorna som kolumner. I skärningspunkterna mellan uppdrag och slingor står det en 1:a om uppdraget ingår i slingan, en 0:a annars. Man kräver vidare att varje rad skall summera till minst det antal personer som behövs för att utföra uppdraget. Uppgiften för optimeringsalgoritmen är nu att ansätta boolska (0/1) värden så att kostnaden minimeras.

Det finns speciella algoritmer för 0/1-matriser som är mycket effektiva och kan hantera stora datamängder.

Steg 1 är inte lika enkelt, och det är här de olika kommersiella lösarna är olika bra. Detta är det steg som benämns *parning*. Det är viktigt att ett bra

urval alternativa slingor genereras i detta steg, eftersom det endast är bland dessa som lösningar sökes.

Slingorna skall dessutom uppfylla villkor som kodar lagar och fackliga avtal, vilka ofta är svåra att representera på ett korrekt och effektivt sätt. Det är långt ifrån alltid klart att avtal och lagstiftning är matematiskt konsistenta och det finns alltid utrymme för tolkningar som ofta varierar lokalt. Eftersom avtal också förändras med tiden, så är det viktigt att de representeras på ett sådant sätt att de är enkla att underhålla.

Vidare representeras ett antal heuristiskt motiverade genereringsvillkor och kostnadsparametrar, vilka begränsar urvalet slingor till de som är "rimliga".

Dessa båda villkorsuppsättningar används sedan i olika metoder för att generera kandidater till slingor för steg 2. Två huvudmetoder används i detta sammanhang: *Heltalsprogrammering* med *Lagrange-relaxering* samt *kolumngenerering* (se t.ex. [DSD84], [RS94]). Detta är fortfarande ett aktivt forskningsområde inom OR.

3.3 Samordningsmekanismer

De flesta mer omfattande verksamheter kräver samordning av flera mer eller mindre heterogena resurser. För riktigt omfattande verksamheter måste dessutom problemet för varje resurs delas upp i mindre delproblem för att dessa skall bli av hanterlig storlek.

Att hantera problem med olika grader av detaljrikedom är nödvändigt om man vill beakta både de kostnader som uppstår lokalt och de vinster man kan göra genom att samordna de lokala verksamheternas användning av delade resurser. Vi behöver en abstrakt (grov) modell för att uppskatta de totala kostnaderna på strategisk och central nivå och konkreta och detaljerade lokala kostnadsmått för varje delproblem. Syftet är att begränsa de lokala valen så att de åtminstone uppfyller elementära krav på den centrala nivån.

3.3.1 Interaktivitet och distribution

Planering av så komplicerade och omfattande verksamheter som de hos en större transporttjänstleverantör måste av flera skäl ske genom manuella beslut i avgörande punkter. Ett beslutstödssystem avser inte att ersätta experten som fattar dessa beslut med en maskin, bara att avlasta denne från rutinmässiga beslut och tillhandahålla ett så bra beslutsunderlag som möjligt.

Många gånger är det också av organisatoriska och politiska skäl bra att decentralisera och distribuera planeringsprocessen. Trots detta finns det också stora vinster att göra i att samordna beslut som tydligt påverkar flera delar av verksamheten. Ett uppenbart exempel är delade resurser som infrastruktur.

Den ansats vi arbetar med ligger i många stycken helt i linje med dessa observationer. Vi arbetar med planeringskomponenter som kan kommunicera lösningsförslag och lokala kostnader och som kan arbeta med olika abstraktionsnivåer. Med denna ansats är det naturligt att en användare inspekterar

resultaten och styr i vilken riktning systemet går vidare, vilka metoder som skall användas etc.

Vidare är varje problemlösningsskomponent implementerad som en separat process (tråd) vilken med primitiver i vårt implementeringsspråk [Smo95], [HvBS98] också kan distribueras till fysiskt skilda platser i ett datornätverk. Detta möjliggör att beslutsprocessen distribueras och att användare med olika ansvarsområden tillsammans men med begränsad kunskap om de övrigas detaljerade modeller och gemensamt enligt överenskomna eller centralt dikterade kriterier bygger upp en fungerande taktisk plan.

De experimentsystem vi arbetar med idag är av naturliga skäl ganska långt ifrån att realisera en sådan vision men vi är övertygade om att denna typ av system med tillräcklig arbetsinsats och expertis i princip är möjliga att ta fram.

3.3.2 Planabstraktion och -konkretion

Ett produktionsplaneringsproblem kan delas upp i flera relativt oberoende dimensioner. En är den uppdelning baserad på resurstyper som skisseras i avsnitt 2.1. Samordning av planerna för dessa resurser kräver uppdrag på flera samtidiga abstraktionsnivåer. Så t.ex. utgör arbetena (tripperna) i spårlägesproblemet uppdragen i fordonsproblemet. Uppdragen (benen) i personalproblemet kan byggas upp av spårlägen men ingår i sig i tripperna. Vid samordning av dessa resursproblem uppstår behovet att på ett enhetligt och entydigt sätt kunna röra sig mellan dessa nivåer.

Vid problemuppdelning av ett sådant resursproblem i mindre delar baserat på t.ex. regioner eller typ av trafik uppstår behovet att jämka samman delplaner genererade lokalt med ofullständig kunskap om de resurskonflikter som uppstår då de skall samordnas. Ett sätt att angripa ett sådant problem är att abstrahera från de givna lösningarna till en gemensam problembeskrivning som bevarar så mycket som möjligt av de lokalt goda egenskaper delplanerna har. För att detta skall vara meningsfullt krävs då naturligtvis att en sådan abstrakt "lösning", som problem betraktat, är enklare att lösa än det problem man skulle fått om man från början slagit samman de ursprungliga problembeskrivningarna.

Ett tredje exempel är de fall då man kan förenkla problemet genom att t.ex. slå samman enkla resurser till mer komplexa för att minska den totala mängden uppdrag i ett givet problem.

I synnerhet i det sista av dessa fall vill man ofta också att de abstraktioner man använder skall uppfylla vissa egenskaper, t.ex. att existensen av en abstrakt lösning *garanterar* existensen av åtminstone en konkret lösning.

Dessa exempel syftar till att illustrera behovet av en entydig och uttömmande formalisering av strukturer och avbildningar som kan användas för att modellera problem av denna karaktär.

Den ansats vi för närvarande arbetar med är en som bygger på ett uppdragsbegrepp som är hierarkiskt organiserat.

Varje primitivt uppdrag och arbete representeras som en UPPDRAGSSTRUKTUR med en entydigt bestämd substruktur. En PROBLEMSTRUKTUR representeras av dels en sådan *uppdragsstruktur*, dels en (partiell) ORDNINGSRRELATION på

de enskilda uppdragen och deras delar och slutligen av en formell representation av de RESURSER uppdragen belägger. ABSTRAKTIONER och KONKRETISERINGAR definieras som avbildningar på problemstrukturer.

Observera att en lösning på ett problem i detta sammanhang kan betraktas som specifik typ av konkretisering av ett problem och i sig utgör en problemstruktur.

I dagens prototypsystem är denna ansats bara delvis genomförd. Ett exempel på en fungerande problemstruktur är de *trippmängder* som utgör den viktigaste komponenten i det protokoll som används för att skicka problem och lösningar mellan komponenterna för fordonsplanering, spårlägesplanering och samordnaren. Abstraktion sker t.ex. genom att fordonskomponenten bortser från mycket av den substruktur som uppstår i spårlägeskomponenten som måste ta hänsyn till samtliga spårresurser och stationsresurser.

3.3.3 Planeringsstrategier

De ovan beskrivna abstraktions- och konkretismekanismerna kan kombineras i komplexa mönster, strategier. Sådana strategier kan användas både för att möjliggöra planering av större problem än vad som är möjligt att hantera som en enhet och för att minska tidsåtgången för planeringen genom att problemet delas upp i mindre delar.

En strategi kan användas t.ex. för att samordna resultaten av flera spårlägesplaneringar. Detta är intressant t.ex. om fjärrtåg går genom ett geografiskt område där även lokal trafik förekommer, och kan även användas för att sammanfoga lösningar på spårlägesplaneringar av geografiskt närliggande områden. Strategier behöver heller inte begränsas till att omfatta abstraktioner och konkretioner, utan kan dessutom användas för att styra samordningen mellan de olika delproblemen.

3.3.4 Planeringsspråk

För att göra det praktiskt möjligt att använda sig av strategier krävs möjlighet att på ett enkelt sätt kunna definiera en strategi. Denna bör dessutom kunna sparas och återanvändas. Detta blir möjligt med hjälp av ett planeringsspråk.

Ett sådant språk bör erbjuda möjligheter att använda mekanismer för abstraktion och konkretion. Det bör dessutom stödja användandet av villkor, så att om en del av en strategi misslyckas kan ett nytt försök göras att lösa denna del av problemet på ett annat sätt. Slutligen behöver en strategi som skrivs i språket kunna lagras och läsas in igen.

Det är värt att notera att ett planeringsspråk inte bara är användbart för lösning av det föreliggande problemet inom transportsektorn, utan kan förenkla lösandet av många slags större planeringsproblem.

4 Tillämpbarhet i andra domäner

De tekniker som beskrivs i denna rapport har tillämpningar i ett brett spektrum av verksamheter i ett modernt samhälle. Nedanstående översikt sammanfattar ett urval av sådana verksamheter.

4.1 Processindustri

4.1.1 Ståltillverkning

Inom stålindustrin förekommer en rad intrikata planeringsproblem kring materialflöden och schemaläggning av bearbetningsenheter. På OVAKO Steel i Hofors utförde SICS 1997-98 inom EU-projektet TACIT en studie av ett bearbetningsflöde omfattande ugnar, valsnings- och kap-operationer. Projektets syfte var att visa på att villkorsteknologin är tillräckligt mogen för att användas för verkliga industriella tillämpningar. Detta problem bestod av 2-3 relativt väl avgränsade resursproblem.

SICS utvecklade i samarbete med planeringsexpertis från OVAKO ett beslutsstödsystem för detta materialflöde. Verktöget läser in data från bearbetningsenheterna och genererar efter ett steg där vissa parametrar kan påverkas manuellt körscheman för ugnar, valsar och viss transportutrustning. Problemet storlek och krav på korta svarstider omöjliggjorde användning av optimerande tekniker. Istället löstes problemet med en *ivrig*¹ algoritm som verkade på en villkorsprogrammeringsmodell. Med denna ansats kunde vi finna planer av hög kvalitet upp till två dygn inom loppet av 30 till 60 sekunder.

Ett avancerat gränssnitt utvecklades för att representera resultaten i form av Gantt-scheman och andra grafiska hjälpmedel. Alternativa planer (baserade på skilda manuella beslut) kan lagras och jämföras manuellt och med avseende på olika kostnadsåtgångar.

4.1.2 Pappersindustri och (bio-)kemisk industri

Inom andra processindustrier, t.ex. inom pappersframställning, petrokemi och bränsleframställning förekommer liknande problem som i OVAKO:s domän. Givetvis förekommer där också andra problem. Ingen så pass komplicerad verksamhet kan modelleras enligt en schablonmodell.

Detta talar för att en teknik som tillåter rika heterogena modeller och i sig kan inkorporera både allmänna och specialiserade problemlösningmetoder för delproblemlösning.

Villkorsprogrammeringen tycks vara en lovande kandidat till en sådan teknik och bör kunna utgöra basen för en allmän metod att modellera en mycket stor klass av problem.

¹Utan söking

4.2 Tillverkningsindustri

TACIT-projektet var ett internationellt samarbete som innefattade också parter också i Frankrike och Ungern. Den ungerska parten utvecklade ett system för schemaläggning av gjutmaskiner för plastkomponenter. I detta fall var problemets storlek sådant att traditionell sökning kunde göras direkt i villkorsprogrammeringssystemet. Systemet är inte optimerande men vissa villkor på kostnadsparametrarna gör att de planer systemet genererar uppnår god kvalitet.

Detta problem liknar många som återfinns i tillverkningsindustrin, t.ex. verktygsindustri, teko, fordonsindustri etc.

4.3 Bygg- och anläggningsindustri

SICS första kontakt med denna typ av problem var inom ramen för förbättring av arbetsmiljö för platschefer m.fl. aktörer på byggarbetsplatser, framför allt husbyggnad. I flera projekt skapades en demonstrationsmiljö som pekade på hur datateknik kan användas för att påtagligt förbättra informationsmöjligheterna och kalkylmöjligheterna för t.ex. platschefer. Problemdomänen karakteriseras av stor konjunkturkänslighet, och ledtiderna har ständigt minskats vilket ställer stor press på aktörerna att klara av snabba omställningar med bibehållande av leveranstidpunkter. Vidare är det vanligt med störningar under själva bygget, dels förändrat beställningsunderlag men också av typen "överraskningar" i markförhållanden, väderpåverkan mm.

Något som växt fram under senare år inom branschen är en standardiserad produktmodell, vilket är en nödvändig komponent för ett datoriserat planeringsstöd. Ritningarna för ett hus ersätts då av en modell i detta standardformat, vilket levereras från arkitekt och tas som indata till ett planeringssystem (eller andra typer av system, t.ex. anbudskalkylsystem). SICS pekade tidigt på behovet av produktmodeller för att skapa en plattform varpå olika hjälpsystem kan implementeras.

Idag används projektplaneringssystem i verksamheten, men en heltäckande koppling från produktdatamodeller som underlag för planeringssystem saknas ännu. Aktivitet pågår dock i branschen att skapa bättre koppling mellan ritningsdata och planeringssystemen, bl.a. Skanska och NCC har program för att åstadkomma bättre datakopplingar mellan sina olika stödsystem.

SICS har under perioden 1988 till 1995 utvecklad demonstrationsprogramvaror, demonstrerat dessa på olika konferenser, hos aktörer och hållit seminarier, samt varit delaktig i utvecklingsprojekt på Skanska (under det s.k. 3T-programmet).

Det franska byggföretaget Bouygues engagerar några av de bästa forskarna inom villkorsprogrammering [CK92], [CL94], [CL95b], [CL95a], [CL96], [CL97], [Cas97].

4.3.1 Transportoptimering

Studier har inom EU-projektet CHIC-II [CHI98] utförts kring problemet att med en given mängd grovt schemalagda uppdrag allokera fordonsresurser utgående från en "egen" fordonspark med låga och förutsägbara kostnader och en mer omfattande "yttre" leverantör av fordonsresurser vars prissättning antas vara högre. Flera alternativa ansatser har tillämpats och jämförts inom projektet.

4.3.2 Masshantering

I ACOOR-projektets styrgrupp har NCCs representanter presenterat ett masshanteringsproblem som uppstår vid väganläggning. En studie av detta problem tycks som en lovande kandidat för att testa de tekniker som tagits fram inom ACOOR på en ganska annorlunda domän. Vi hoppas att i diskussioner med NCC kunna närma oss detta problem.

4.4 Transporttjänstproduktion

4.4.1 Spårbundna transporter

Spårbundna transporter innefattar mycket fler problemställningar än de som vi specifikt arbetat med inom ACOOR och TUFF-projekten. Ett exempel är flödesoptimering av vagnresurser där OR-tekniker framgångsrikt tillämpats [KL96], [Lob98], [Job98]. Andra problem innefattar t.ex. att generera trippspecifikationer givet ett statistiskt flödesmått. Dessa problem skiljer sig sannolikt mellan gods och persontransporter. Sådana analyser kan användas vid anbudsförfaranden.

4.4.2 Vägtransporter

Så kallade resandeproblem hör till de svåraste och samtidigt bäst studerade områdena inom teorin för algoritmer. Dessa problem är i allmänhet mycket svårlösta men många specifika problem har framgångsrikt angripits med både OR- och villkorstekniker liksom lokal sökning och specialiserade heuristiker. Exempel är allokering av fordonsresurser för att utföra givna punkt-till-punkt-transporter från en eller flera depåer. Om problemet också innefattar att fastställa exakta tider inom fixerade tidsfönster är problemet tekniskt mycket likt lokomloppsproblemet i ACOOR.

4.4.3 Allmänna persontransportsystem

Liknande problem förekommer i planering av fordons- och personalresurser för allmänna transportsystem som tunnelbanor, spårvagnar och bussar. Även dessa problem är i renodlad form väl studerade inom OR men också här förekommer ibland komplexa bivillkor och samordningsproblem som OR-teknikerna är illa lämpade att hantera.

4.4.4 Flygtrafikplanering

Inom flygtrafikplanering används i förhållandevis hög grad optimerande system för att belägga de flygande fordons- och personalresurserna. Det Göteborgsbaserade företaget Carmen Systems är marknadsledande inom personalplaneringssystem för flyget och är aktiva också inom järnvägsindustrin. Andra problem är mindre väl utforskade, t.ex. minimering av bränslekostnader med hänsyn tagen till lokala väderförhållande och sent gjorda vägval.

4.4.5 Samordning av materialflöden i JIT-system

Samordningsproblem av liknande karaktär som de som förekommer i järnvägsindustrin förekommer också då flera mer eller mindre självständiga produktionsenheter skall samverka i produktionen av komplexa produkter. Sådan problem är välkända från t.ex. bilindustrin. Problemet är att i möjligaste mån minska lagerhållning av dyrbara halvfabrikat och komponenter. Metoden att köpa komponenter från underleverantörer och hålla minimala lager minskar drastiskt kostnaden för produktion men tenderar också att öka systemen störningskänslighet. En viss samordning av produktionsplaner är därför ofta nödvändig för att upprätthålla leveranssäkerhet.

4.5 Data- och Telekommunikation

Inom området för planering av utbyggnad av och resursanvändning i kommunikationsnätverk förekommer en rad optimeringsproblem som kan angripas med de tekniker som diskuterats i denna översikt. Se t.ex. [FB97], [BDD99]. Detta är ett stort område som vi f.n. inte har stor kunskap om men förefaller mycket lovande som tillämpningsområde för dess tekniker.

5 Slutsatser

Den modellering av produktionsplaneringsproblemet i järnvägsdomänen har resulterat i utforskandet av en rad tekniker och metoder som är användbara även utanför detta begränsade problemområde.

Referenser

- [AC91] D. Applegate and W. Cook. A computational study of the job-shop scheduling problem. *ORSA Journal of Computing*, 3(2):149–156, 1991.
- [AMO93] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows*. Prentice Hall, 1993.
- [BC94] N. Beldiceanu and E. Contejean. Introducing global constraints in CHIP. *Mathematical computer modeling*, 20(12):97–123, 1994.

- [BDD99] P. Boizumault, P. David, and H. Djelab. A repair algorithm for allocating resources in mobile telephone network. In *Proceedings of the conference on Practical Applications of Constraint Technology and Logic Programming (PACLP)*, 1999.
- [Bel00] Nicolas Beldiceanu. Global constraints as graph properties on structured networks of elementary constraints of the same type. Research R:00:01, SICS, 02 2000.
- [BG81] L. Bodin and B. Golden. Classification in vehicle routing and scheduling. *Networks*, 11(97–108), 1981.
- [BLP95] P. Baptiste and C. Le Pape. A theoretical and experimental comparison of constraint propagation techniques for disjunctive scheduling. In *Proceedings of the fourteenth international joint conference on artificial intelligence*, pages 400–606, Montreal, Quebec, 1995.
- [BWZ97] M. R. Bussieck, T. Winter, and U. T. Zimmermann. Discrete optimization in public rail transport. *Mathematical Programming*, 79:415–444, 1997.
- [Cas97] Y. Caseau. Using constraint propagation for complex scheduling problems: managing size, complex resources and travel. In G. Smolka, editor, *CP'97 – Principles and Practice of Constraint Programming*, volume 1330 of *LNCS*, Linz, Austria, Oct/Nov 1997. Springer-Verlag.
- [CHI98] CHIC-II partners. Homepage for the chic-ii project with on line report archive. Web, 1998.
- [CK92] Y. Caseau and P. Kopperstein. A cooperative-architecture expert system for solving large time/travel assignment problems. In *Proceedings of DEXA '92 (International Conference on Databases and Expert Systems Applications)*, 1992.
- [CL94] Y. Caseau and F. Laburthe. Improved CLP scheduling with task intervals. In P. van Hentenryck, editor, *Proceedings of the eleventh International Conference on Logic Programming*, volume 78, Santa Margherita Ligure, Italy, 1994. MIT Press.
- [CL95a] Y. Caseau and F. Laburthe. Cumulative scheduling with task intervals. Technical report, Laboratoire d'Informatique de l'Ecole Normale Supérieure LIENS, Département de Mathématiques ed d'Informatique, 45 rue d'Ulm, 75232 Paris Cedex 05, France, 1995.
- [CL95b] Y. Caseau and F. Laburthe. Disjunctive scheduling with task intervals. Technical Report 95-25, Laboratoire d'Informatique de l'Ecole Normale Supérieure LIENS, Département de Mathématiques ed d'Informatique, 45 rue d'Ulm, 75232 Paris Cedex 05, France, 1995.

- [CL96] Y. Caseau and F. Laburthe. Improving branch and bound for jobshop scheduling with constraint propagation. Technical report, Laboratoire d'Informatique de l'Ecole Normale Supérieure LIENS, Département de Mathématiques et d'Informatique, 45 rue d'Ulm, 75232 Paris Cedex 05, France, 1996.
- [CL97] Y. Caseau and F. Laburthe. Solving small TSP with constraints. In L. Naish, editor, *Proceedings of the Joint International Conference and Symposium on Logic Programming*. MIT Press, 1997.
- [CP89] J. Carlier and E. Pinson. An algorithm for solving the job-shop problem. *Management Science*, 35(2):164–176, 1989.
- [CP94] J. Carlier and E. Pinson. Adjustments of heads and tails for the job-shop scheduling problem. *European Journal of Operational Research*, 78:146–161, 1994.
- [DHKK97] J. Drott, E. Hasselberg, N. Kohl, and M. Kremer. A planning system for locomotive scheduling. Technical report, Swedish State Railways, Stab Tågplanering, Stockholm, Sweden, and Carmen Systems AB, Jul 1997.
- [DSD84] J. Desrosiers, F. Soumis, and M. Desrochers. Routing with time windows by column generation. *Networks*, 14:545–565, 1984.
- [FB97] Tom Früwirth and P. Brisset. Optimal planning of digital cordless telecommunications. In *Proceedings of the conference of Practical Applications of Constraint Technology (PACT)*, 1997.
- [GA86] B. L. Golden and A. A. Assad. Vehicle routing with time-window constraints: Algorithmic solutions. *American Journal of Mathematical and Management Sciences*, 6, 1986.
- [Gos93] V. Gosselin. Train scheduling using constraint programming techniques. In *13th conference on AI, expert systems and natural language*, Avignon, 1993.
- [HvBS98] Seif Haridi, Peter van Roy, Per Brand, and Christian Schulte. Programming languages for distributed applications. *New Generation Computing*, 16:223–261, 1998.
- [Job98] M. Joborn. Empty freight car distribution at swedish state railways. Lic. thesis, Department of Mathematics, University of Lindköping, Sweden, 1998.
- [KL96] A. Kokott and A. Löbel. Lagrangian relaxations and sub-gradient methods for multiple-vehicle scheduling problems. Preprint SC 96-22, Konrad-Zuse-Zentrum für Informationstechnik, Berlin, 1996.

- [Löb98] A. Löbel. *Optimal Vehicle Scheduling in Public Transit*. PhD thesis, TU Berlin, 1998. Shaker-Verlag, Aachen.
- [MS94] P. Martin and D. B. Shmoys. A new approach to computing optimal schedules for the job-shop scheduling problem. pages 389–403, 1994.
- [PR93] J.-Y. Potvin and J.-M. Rousseau. A parallel route building algorithm for the vehicle routing and scheduling problem with time windows. *European Journal of Operational Research*, 66:331–340, 1993.
- [Ree95a] C. Reeves, editor. *Modern Heuristic Techniques for Combinatorial Optimization*, chapter Chapter 6: Lagrangean Relaxation. McGraw-Hill International (UK) Ltd., 1995.
- [Ree95b] C. Reeves, editor. *Modern Heuristic Techniques for Combinatorial Optimization*. McGraw-Hill International (UK) Ltd., 1995.
- [RS94] C. C. Ribeiro and F. Soumis. A columns generation approach to the multiple-depot vehicle scheduling problem. *Operations Research*, 42(1):41–52, 1994.
- [Sav85] M. W. P. Savelsbergh. Local search in routing problems with time windows. *Annals of Operations Research*, 4:285–305, 1985.
- [SD88] M. M. Solomon and J. Desrosiers. Time window constrained routing and scheduling problems. *Transportation Science*, 22(1):1–13, 1988.
- [Sim96] H. Simonis. Constraint logic programming. Technical report, COSYTEC SA, 1996.
- [Smo95] G. Smolka. The Oz programming model. In J. van Leeuwen, editor, *Computer science today*, volume 1000 of *Lecture notes in computer science*, pages 324–343. Springer verlag, 1995.
- [Sol86] M. M. Solomon. On the worst-case performance of some heuristics for the vehicle routing and scheduling problem with time window constraints. *Networks*, 16:161–174, 1986.
- [Sol87] M. M. Solomon. Algorithms for the vehicle routing and scheduling problem with time window constraints. *Operations Research*, 35(2):254–265, March-April 1987.
- [TP93] P.M. Thompson and H.N. Psaraftis. Cyclic transfer algorithms for multi-vehicle routing and scheduling problems. *Operations Research*, 41(5):935–946, 1993.
- [Tsa93] E. Tsang. *Foundations of Constraint Satisfaction*. Academic Press, 1993.
- [vH89] P. van Hentenryck. *Constraint satisfaction in logic programming*. Programming logic series. The MIT press, Cambridge, MA, 1989.

