

Optimering av scheman för en verklig produktionscell: tidsdiskretisering reducerar lösningstiden utan att lösningarnas kvalitet försämras

Karin Thörnblad^{1a}, Torgny Almgren^{1b}, Ann-Brith Strömberg^{2a}
och Michael Patriksson^{2b}

1. *Chalmers Tekniska Högskola, Inst. för Matematiska Vetenskaper
och Volvo Aero Corporation, Logistikutveckling*

461 81 Trollhättan

^akarin.thornblad@volvo.com, ^btorgny.almgren@volvo.com

2. *Chalmers Tekniska Högskola och Göteborgs Universitet
Inst. för Matematiska Vetenskaper*

SE-412 96 Göteborg

^aanstr@chalmers.se, ^bmipat@chalmers.se

Sammanfattning

Vid optimering av industriella schemalägningsproblem är kompromissen mellan beräkningstid och resultatnivå ofta kritisk, med risken att en optimeringsmodell förenklas så mycket att dess resultat inte är praktiskt relevant. I artikeln jämförs två matematiska optimeringsmodeller för optimal schemaläggning av en verklig produktionscell på Volvo Aero. I den ena modellen beskrivs bl. a. operationers starttider m.h.a. kontinuerliga variabler, medan tiden har diskretiserats i den andra modellen. Diskretiseringen innebär att planeringshorisonten indelas i ett antal tidssteg, här kallade diskreta tidsperioder. Det är en förenkling eftersom all indata då måste uttryckas i helt antal diskreta tidsperioder och varje operation endast kan schemaläggas vid början av en tidsperiod. Den senare modellen visade sig överlägsen vad gäller beräkningstider och bedömdes därför ha mycket stor potential, trots risken för approximeringsfel p.g.a. tidsdiskretiseringen.

I artikeln presenteras tester med verkliga indata, vilka visar att approximeringsfelen blir mycket små när den diskreta tidsperiodens längd är en timma eller mindre, trots att några operationstider är kortare än så. Det optimala schemat från den tidsdiskreta modellen har före jämförelsen justerats så att alla operationstider är de ursprungliga, detta utan att förändra vare sig maskinval eller valda sekvenser på maskinerna. Denna modell har goda förutsättningar att fungera praktiskt, eftersom den både är snabb och ger goda resultat.

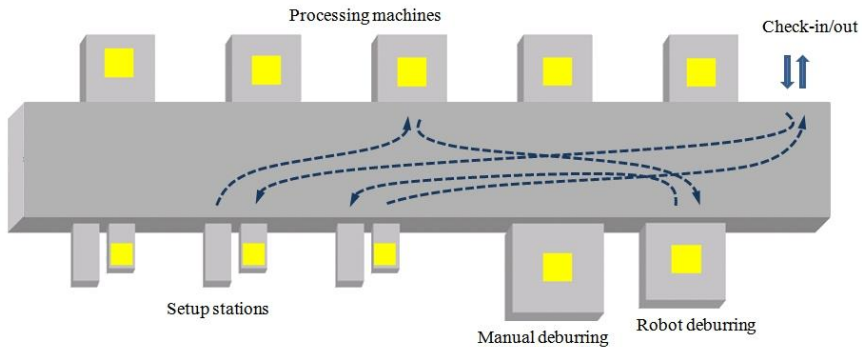
Nyckelord: Matematisk optimering, tidsdiskretisering, flexible job shop

1. Introduktion

En av Volvo Aeros senaste stora investeringar kallas "multitaskcellen", en produktionscell innehållande tio resurser. Cellen byggdes med målen att uppnå en högre grad av maskinutnyttjande, kortare produktledtider och högre flexibilitet avseende både produktmix och bearbetningstyp.

1.1. Problembeskrivning

Multitaskcellen kan utföra en mängd olika operationer eftersom fem av cellens resurser består av så kallade multitaskmaskiner som kan bearbeta materialet på tre olika sätt: fräsning, svarvning och borrar. Vanligtvis besöker varje produkt cellen flera gånger innan den är färdigbearbetad. Inne i cellen följer varje enskild detalj en specifik rutt bestående av tre till fem operationer, med början och slut i en av de tre "set up"-stationerna, där montering i och borttagning av fixturer sker. Den andra operationen i ruten är alltid bearbetning i en av de fem multitaskmaskinerna. Vissa detaljer behöver manuell och/eller robot-gradning (DBR). Figur 1 illustrerar en möjlig väg för en detalj genom multitaskcellen.



Figur 1: Översikt över multitaskcellen. De streckade linjerna indikerar en möjlig väg för en detalj genom multitaskcellen.

2. Litteraturstudie

En utbredd användning av fleroptionsmaskiner i moderna verkstäder har gett upphov till ett nytt schemalägningsproblem att studera vid sidan av det klassiska ”jobshop”-problemet, som innebär att n jobb ska schemaläggas på m maskiner och varje jobb har en fördefinierad rutt mellan maskinerna samt varje operation kan utföras av endast en maskin (Brucker, 2007). Det så kallade ”flexible jobshop”-problemet (FJSP) är en utvidgning av ”jobshop”-problemet, i den meningen att varje operation kan ha mer än ett maskinalternativ (Baykasoglu och Özbakir, 2010). Problemet blir därmed ännu mer komplext eftersom - förutom att schemalägga alla jobb - även ett tilldelningsproblem måste lösas, där jobben fördelas mellan tillåtna maskiner på ett optimalt sätt.

FJSP är NP-svårt liksom jobshop-problemet för instanser då antalet maskiner är fler än två (Brucker m. fl., 1997). Med NP-svårt menas att det inte existerar någon känd algoritm som inom polynomiell tid kan lösa alla tänkbara fall. Med andra ord riskerar lösningstiden att öka exponentiellt med antalet jobb. Svårigheten under 1900-talet att med hjälp av matematisk optimering hitta lösningar även till relativt små jobshop-problem, med exempelvis 10 jobb och 10 maskiner, har bidragit till att mängder av approximativa algoritmer har utvecklats för att ta fram möjliga lösningar för jobshop-problemet och FJSP (Jain and Meeran 1999).

Utvecklingen av optimeringsmodellering, -metoder och -programvaror har tillsammans med utvecklingen av datorhårdvara under de senaste årtiondena

dock inneburit att medelstora problem numera kan lösas med hjälp av matematisk modellering och optimeringsmetoder. Fördelen med matematisk optimering är att lösningarna som erhålles antingen är optimala, eller redovisas tillsammans med ett intervall (kallat gap) inom vilket det optimala värdet på målfunktionen garanterat återfinns.

En av de första formuleringarna av ett schemalägningsproblem genom att nyttja linjär heltalsoptimering gjordes av (Manne 1960), då han formulerade en matematisk modell av problemet att optimalt schemalägga ett antal jobb, med så kallade ”precedence constraints”, i en enda maskin. Med ”precedence constraints” menas villkor som reglerar ordningsföljden mellan jobben. Manne använde sig av kontinuerliga tidsvariabler för jobbets starttider, samt binära beslutsvariabler för ordningen mellan jobben. Eftersom detta sätt att modellera ett schemalägningsproblem är tämligen intuitivt, återfinns liknande modeller för jobshop-problemet i läroböcker för optimering (Taha 2007) samt även i nyskrivna forskningsartiklar (Baykasoglu 2010).

Ett annat sätt att formulera jobshop-problem enligt linjär heltalsoptimering eller ”Mixed Integer Linear Programming” (MILP) är att dela in tiden i diskreta tidssteg och skapa en binär variabel för varje jobb och tidpunkt t vilken antar värdet 1 om jobbet ska startas under den tidsperiod som börjar vid tiden t och 0 annars (van den Akker 2000).

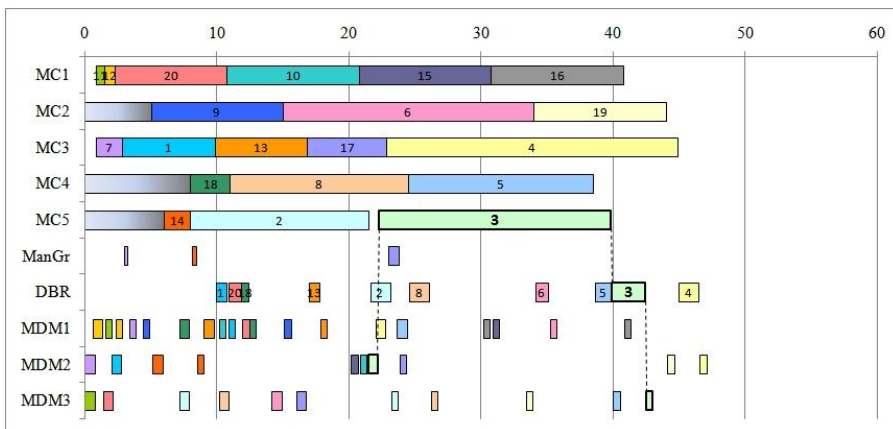
3. Matematiska formuleringar

I detta avsnitt presenteras två modeller för att ta fram optimala scheman för multitaskcellen; den ena kallas ingenjörsmoellen och tillhör den familj av modeller som har sitt ursprung i (Manne 1960); den andra är en tidsdiskret modell inspirerad av (Wolsey 1997). Problemet att ta fram optimala scheman för multitaskcellen är ett FJSP och vår tidsdiskreta modell är, enligt vår kännedom, den första av sitt slag som utvecklats för en ”flexible jobshop”.

3.1. Uppdelning av problemet

Den första modellen som utvecklades för multitaskcellen tillhörde ”Manne-familjen” och hade alltför långa beräkningstider vid tester med verkliga instanser. Därför delades problemet upp i två delar enligt följande: först bestäms det optimala schemat för de operationer som ska utföras i de fem multitaskmaskinerna, därefter skapas ett körbart schema för de övriga operationerna montering i fixtur, manuell gradning, robotgradning och demontering ur fixtur. Anledningen till att problemet delas upp på just detta sätt

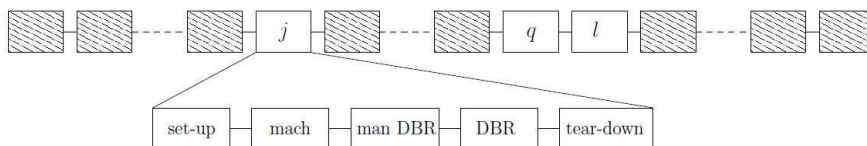
är att beläggningen är väsentligt högre i multitaskmaskinerna än i övriga resurser i produktionscellen; se Figur 2. Det första problemet att bestämma det optimala schemat för multitaskmaskinerna kallas ”maskinproblemet”, och det andra problemet kallas ”feasibility-problemet”.



Figur 2: Ett exempel på schema för multitaskcellen. Man ser tydligt att beläggningen för de fem multitaskmaskinerna, här betecknade MC1-5, är betydligt högre än för övriga resurser. Rutten för jobb nr 3 är markerad med streckade linjer.

3.2. Definition av parametrar och mängder

Mängden \mathcal{K} betecknar resurserna, d.v.s. de fem multitaskmaskinerna. De jobb som ska schemaläggas betecknas med mängden \mathcal{J} . En del jobb är knutna till samma fysiska detalj och alla par av sådana jobb som ligger efter varandra i operationsflödet utgör mängden \mathcal{Q} . För den detalj vars operationsflöde är beskriven i Figur 3, tillhör t ex paren (j,q) och (q,l) mängden \mathcal{Q} .



Figur 3: En detaljs operationsflöde med jobben j , q och l som ska utföras i multitaskcellen. De skuggade rutorna betecknar operationer som ska utföras utanför multitaskcellen. Paren (j,q) och (q,l) tillhör mängden \mathcal{Q} .

För varje jobb j är följande data givna:

- en processtid p_j ,
- en tidpunkt då detaljen tidigast är tillgänglig r_j (release date) samt
- en färdigtidpunkt d_j (due date).

Eftersom en del jobb får bearbetas endast i en delmängd av multitaskmaskinerna har en parameter λ_{jk} skapats. Parametern λ_{jk} antar värdet 1 om jobb j kan bearbetas i resurs k och den antar värdet 0 ifall jobb j ej kan bearbetas i resurs k . Tidpunkten då en resurs är tillgänglig för första gången betecknas a_k . För varje par (j, q) i mängden \mathcal{Q} , betecknar v_{jq} den planerade leddtiden mellan sluttiden för jobb j och starttiden för maskinoperationen för jobb q .

3.3. Den tidsdiskreta modellen

Tiden från starten för schemat tills alla jobb med säkerhet schemalagts delas in i diskreta tidssteg, se Figur 4. Längden på de diskreta perioderna betecknas med Δ



Figur 4: Planeringshorisonten delas in i $T+1$ diskreta tidsperioder.

För att formulera den tidsdiskreta modellen krävs en binär beslutsvariabel x_{jku} som antar värdet 1 om jobb j startar i resurs k vid starten av tidsperiod u och 0 annars. Sluttiden för jobb j , en kontinuerlig variabel som betecknas s_j , kan uttryckas som en funktion av variablerna x_{jku} (villkor (1g) nedan). En positiv förseningen (på eng. begreppet "Tardiness") av jobb j definieras som $h_j = \max\{0, s_j + p_j^{\text{pm}} - d_j\}$, $j \in \mathcal{J}$, och utgör en del av målfunktionen.

Målet i maskinproblemet är att minimera summan av de positiva förseningarna, men för att också kunna prioritera jobb som inte är försenade, d.v.s. har noll försening, har även summan av sluttiderna adderats till målfunktionen, så att *alla* jobb schemaläggs så tidigt som möjligt.

I modellen nedan definieras $(\xi)_+ := \max\{0, \xi\}$ samt $v_{jq}^{\text{pm}} := p_j + p_j^{\text{pm}} + v_{jq}$. Den tidsdiskreta modellen av maskinproblemet är formulerad enligt

$$\text{minimera} \quad \sum_{j \in \mathcal{J}} (s_j + h_j), \quad (1a)$$

$$\text{då} \quad \sum_{k \in \mathcal{K}} \sum_{u \in \mathcal{T}} x_{jku} = 1, \quad j \in \mathcal{J}, \quad (1b)$$

$$\sum_{u \in \mathcal{T}} x_{jku} \leq \lambda_{jk}, \quad j \in \mathcal{J}, k \in \mathcal{K} \quad (1c)$$

$$\sum_{j \in \mathcal{J}} \sum_{\nu=(u-p_j+1)_+}^u x_{jk\nu} \leq 1, \quad k \in \mathcal{K}, u = 0, \dots, T, \quad (1d)$$

$$\sum_{k \in \mathcal{K}} \left(\sum_{\mu=0}^u x_{jk\mu} - \sum_{\nu=0}^{u+v_{jq}^{\text{pm}}} x_{qk\nu} \right) \geq 0, \quad (j, q) \in \mathcal{Q}, u = 0, \dots, T - v_{jq}^{\text{pm}}, \quad (1e)$$

$$x_{jku} = 0, \quad (j, q) \in \mathcal{Q}, k \in \mathcal{K}, u = T - v_{jq}^{\text{pm}}, \dots, T, \quad (1f)$$

$$\sum_{k \in \mathcal{K}} \sum_{u \in \mathcal{T}} u x_{jku} + p_j^{\text{pm}} = s_j, \quad j \in \mathcal{J}, \quad (1g)$$

$$s_j - h_j \leq d_j, \quad j \in \mathcal{J}, \quad (1h)$$

$$h_j \geq 0, \quad j \in \mathcal{J}, \quad (1i)$$

$$x_{jku} = 0, \quad (1j)$$

$$j \in \mathcal{J}, k \in \mathcal{K}, u = 0, 1, \dots, \max\{r_j, a_k\},$$

$$x_{jku} \in \{0, 1\}, j \in \mathcal{J}, k \in \mathcal{K}, u \in \mathcal{T}. \quad (1k)$$

Villkor (1b) ser till att varje jobb blir schemalagd exakt en gång och villkor (1c) reglerar att varje jobb schemaläggs i en tillåten resurs. Villkor (1d) säkerställer att inget annat jobb startas i resurs k under tiden jobb j bearbetas. Villkoren (1e) och (1f) säkerställer att jobb j och q , där $(j, q) \in \mathcal{Q}$, schemaläggs i rätt ordning och att minst tiden v_{jq}^{pm} har förflutit sedan starttiden för jobb j innan jobb q startas.

Villkoren (1g)-(1i) definierar sluttider och positiv försening, vilka utnyttjas i målfunktionen (1a). Villkor (1j) reglerar jobbets starttider, så att inget jobb schemaläggs innan tiden r_j har förflutit eller innan resursen blivit tillgänglig för första gången.

3.4. Ingenjörsmodellen

Den första modellen som utvecklades för maskinproblemet kallas ingenjörsmodellen eftersom variabler och villkor är intuitivt formulerade. Modellen är presenterad mer i detalj i (Thörnblad m fl 2010). Den utnyttjar två grupper av binära beslutsvariabler som kopplar samman jobb med resurser samt reglerar ordningen mellan jobben på varje maskin. Den första gruppen av

variabler betecknas z_{jk} som får värdet 1 då jobb j ska bearbetas i resurs k , 0 annars. Den andra gruppen betecknas y_{jqk} och får värdet 1 om jobb j föregår jobb q och båda ska bearbetas i resurs k , 0 annars. Vidare används liksom för den tidsdiskreta modellen, variablerna s_j för sluttider och h_j för positiv försening samt en kontinuerlig tidsvariabel t_j för jobbets starttider.

Målfunktionen är densamma som för den tidsdiskreta modellen och problemet att schemalägga multitaskcellen formuleras därför som det att

$$\text{minimera} \quad \sum_{j \in \mathcal{J}} (s_j + h_j), \quad (2a)$$

$$\text{då} \quad \sum_{k \in \mathcal{K}} z_{jk} = 1, \quad j \in \mathcal{J}, \quad (2b)$$

$$z_{jk} \leq \lambda_{jk}, \quad j \in \mathcal{J}, k \in \mathcal{K}, \quad (2c)$$

$$y_{jqk} + y_{qjk} \leq z_{jk}, \quad j, q \in \mathcal{J}, k \in \mathcal{K}, j \neq q, \quad (2d)$$

$$y_{jqk} + y_{qjk} + 1 \geq z_{jk} + z_{qk}, \quad j, q \in \mathcal{J}, k \in \mathcal{K}, j \neq q, \quad (2e)$$

$$t_j + p_j - t_q \leq M(1 - y_{jqk}), \quad j, q \in \mathcal{J}, k \in \mathcal{K}, j \neq q, \quad (2f)$$

$$t_j \geq r_j, \quad j \in \mathcal{J}, \quad (2g)$$

$$t_j \geq a_k z_{jk}, \quad j \in \mathcal{J}, \quad (2h)$$

$$t_q \geq s_j + v_{jq}, \quad (j, q) \in \mathcal{Q}, \quad (2i)$$

$$s_j - t_j = p_j + p_j^{\text{pm}}, \quad j \in \mathcal{J}, \quad (2j)$$

$$h_j \geq s_j - d_j, \quad j \in \mathcal{J}, \quad (2k)$$

$$h_j \geq 0, \quad j \in \mathcal{J}, \quad (2l)$$

$$t_j \geq 0, \quad j \in \mathcal{J}, \quad (2m)$$

$$z_{jk} \in \{0, 1\}, \quad j \in \mathcal{J}, k \in \mathcal{K}, \quad (2n)$$

$$y_{jqk} \in \{0, 1\}, \quad j, q \in \mathcal{J}, k \in \mathcal{K}, j \neq q, \quad (2o)$$

där p_j^{pm} är summan av processtiderna för resterande operationer i multitaskcellen efter maskinbearbetningen för jobb j .

4. Beräkningsresultat

De två modellerna presenterade ovan är ekvivalenta endast ifall all indata ges i hela multiplar av den diskreta tidsperiodens längd \mathcal{L} . Det finns en risk för approximeringsfel vid användande av den tidsdiskreta modellen eftersom de verkliga processtiderna är givna med 0,1 h noggrannhet vilket är ett för litet

intervall. Nedan beskrivs resultat erhållna från beräkningar med den tidsdiskreta modellen med verkliga data från Volvo Aero för ett antal valda värden på längden av den diskreta tidsperioden. Dessa resultat jämförs även med resultaten från beräkningar med ingenjörsmodellen avseende både noggrannhet och beräkningstid.

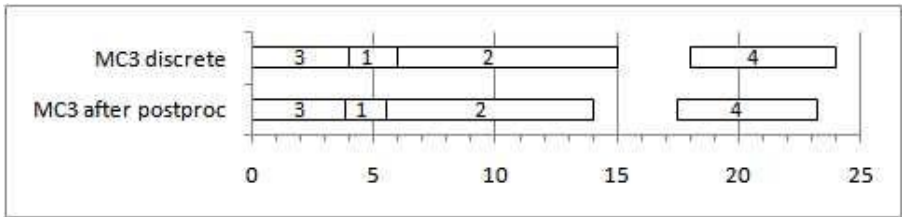
4.1. Indata

För närvarande bearbetas åtta olika produkter med ca 30 olika maskinoperationer i multitaskcellen. Sex verkliga scenarier samlades in från multitaskcellen under hösten 2010. I varje scenario ordnades jobben efter stigande värde på r_j (den tidpunkt då detaljerna tidigast är tillgängliga för bearbetning, ”release date”). Därefter skapades olika testproblem genom att kapa kön av jobb vid ett bestämt antal, så att det minsta problemet består av fem st. jobb och det största av 70 st. jobb.

Eftersom det finns en variabel för varje jobb, tidpunkt och resurs för den tidsdiskreta modellen beror antalet variabler på valet av längden av den diskreta tidsperioden \mathcal{L} . Dessutom beror antalet villkor i grupperna (1d) och (1e) av antalet diskreta tidssteg. Därför beror både krav på minnesutrymme och beräkningstid på valet av \mathcal{L} och valet av T , d.v.s. längden på planeringshorisonten. Ett lämpligt värde på T räknas ut med hjälp av en heuristik, som snabbt hittar ett bra körbart schema. Även parametern M i ingenjörsmodellen bestäms med hjälp av denna heuristik.

4.2. Bearbetning av resultat före jämförelse

Start- och sluttiderna för operationer som erhålls av den tidsdiskreta modellen ges i hela multiplar av den diskreta tidsperiodens längd. Därför skiljer sig de optimala värdena på målfunktionerna (1a) och (2a). Med anledning härav har sluttiderna räknats om i efterhand genom att nyttja den verkliga (icke-diskreta) datan samtidigt som jobbets ordningsföljd på varje maskin bibehållits från resultatet av beräkningen med den tidsdiskreta modellen. I figur 5 är t ex det diskreta värdet på r_j 18h, medan det i verkligheten är 17,5h, vilket är anledningen till förändringen av pausen mellan jobb 2 och jobb 4 i schemat. En dylik efterbehandling av resultaten från den tidsdiskreta modellen tar maximalt 0.04 s för ett problem med 70 jobb och är därför försumbar i sammanhanget.

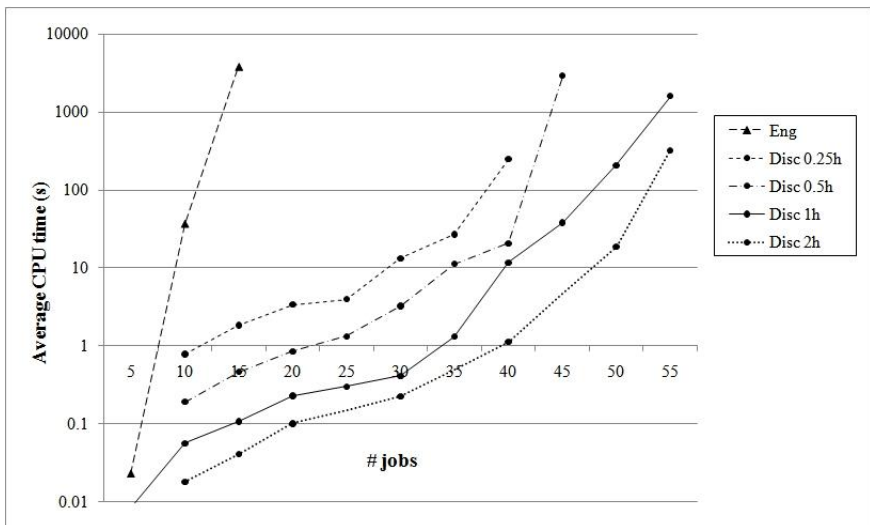


Figur 5: Resultaten från den tidsdiskreta modellen är efterbearbetad med den verkliga icke-diskreta datan.

4.3. Resultat

De efterbearbetade resultaten från den tidsdiskreta modellen (1) har jämförts med resultat från ingenjörsmodellen (2). Optimeringsmjukvaran som nyttjades var AMPL-CPLEX12 och exekverades i en beräkningsdator med två st. 2.66GHz Intel Xeon 5650-processorer, med totalt 24 trådar och 48Gbyte RAM-minne.

I Figur 7 visas medelvärdet av beräkningstiden (CPU-tid) för de sex scenarierna för varje modell och problemstorlek.



Figur 7: Medelvärde av beräkningstid i sekunder för de sex verkliga scenarierna som funktion av modell och problemstorlek. Endast de fall där resultat erhållits för alla sex scenarier har tagits med. Lägg märke till att skalan på den vertikala axeln är logaritmisk. Eng = Ingenjörsmodellen, Disc = tidsdiskreta modellen med varierande värden på längden, ℓ , av den diskreta tidsperioden.

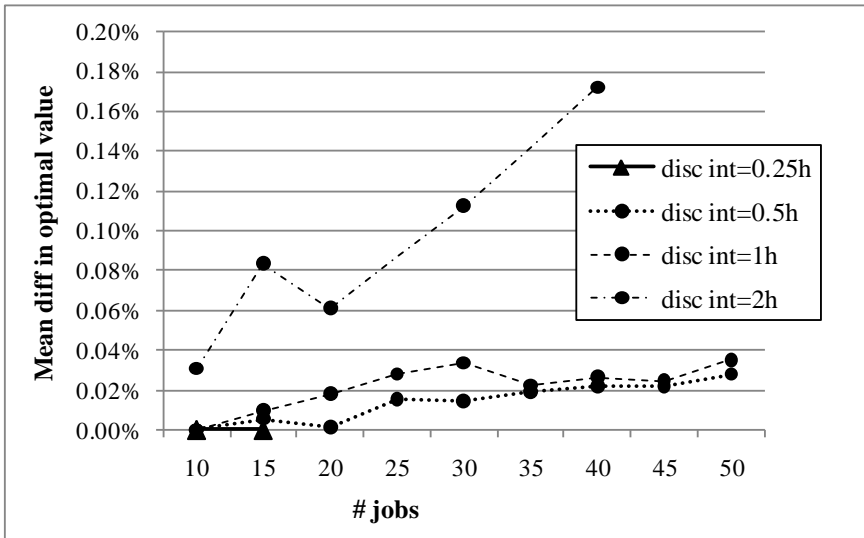
Ingenjörsmodellen har klart längre beräkningstider redan för små problem, och vissa problem med 20 st. jobb avbröts p.g.a. minnesproblem alternativt alltför långa beräkningstider. Den tidsdiskreta modellen klarar däremot att finna optimala värdet på målfunktionen för betydligt större problem. Det finns ca 30 lagerplatser i multitaskcellen, vilket innebär att det skulle kunna finnas 30 detaljer med $r_j = 0h$, vilka borde beaktas simultant. Eftersom en del detaljer som är på väg till multitaskcellen antagligen är nära och har ett litet värde på r_j är det realistiskt att vilja skapa ett detaljerat schema för ca 45 detaljer. Den tidsdiskreta modellen med $\ell = 1h$ krävde i genomsnitt 40 CPU-sekunder för att finna en optimal lösning för scenarier med 45 jobb.

I Tabell 1 återfinns medelvärdena av de relativa skillnaderna mellan de olika modellernas målfunktionsvärden. Den diskreta modellen (1) med $\ell = 0.25h$ har enbart jämförts med ingenjörsmodellen, och det finns ingen skillnad mellan de optimala värdena för dessa två modeller. Eftersom det endast finns resultat upp till problem med 15 jobb med ingenjörsmodellen, har endast problemen med 10 respektive 15 jobb kunnat jämföras. De andra testerna med högre värden på ℓ har jämförts mot optimala målfunktionsvärdet från beräkningen med $\ell = 0.25h$. Felet ökar med antal jobb för modellen med $\ell = 2h$, medan resultatet från övriga modeller ligger nära varandra. När den diskreta tidsperiodens längd är kortare än 1h påverkas alltså resultatet mycket litet, trots att de kortaste operationstiderna är under 1h. Resultatet i Tabell 1 visas i diagramform i Figur 8.

Vid en snabb betraktelse förefaller skillnaderna även mellan modellen med $\ell = 2h$ som mycket små jämfört med modellen med $\ell = 0.25h$, men det rör sig trots allt om ett antal timmar eftersom förseningstimmarna i en del fall var många. Dock skulle den tidsdiskreta modellen (1) med $\ell = 2h$ kunna fungera som ett verktyg för en grovre långsiktig planering. Nämnas kan också att den relativa differensen mellan de resulterande målfunktionsvärdena från prioriteringsregeln "Earliest Due Date" (EDD) och ingenjörsmodellen var större än 10% (Thörnblad och Kjellsdotter 2010).

Tabell 1: Jämförelse av modellerna. Den diskreta modellen (1) med $\ell = 0.25h$ har jämförts med ingenjörsmodellen (2), medan den diskreta modellen med $\ell > 0.25h$ har jämförts med samma modell med $\ell = 0.25h$.

# jobs	Diff to (2) $\ell = 0.25h$ (%)	Diff to (1) $\ell = 0.5h$ (%)	Diff to (1) $\ell = 1h$ (%)	Diff to (1) $\ell = 2h$ (%)
10	0.000	0.000	0.000	0.031
15	0.000	0.005	0.010	0.083
20	Resultat saknas	0.001	0.018	0.061
25	Resultat saknas	0.015	0.028	
30	Resultat saknas	0.015	0.034	0.112
35	Resultat saknas	0.019	0.022	
40	Resultat saknas	0.022	0.027	0.172
45	Resultat saknas	0.022	0.025	
50	Resultat saknas	0.028	0.035	0.335
Mean	0.000	0.014	0.022	0.132



Figur 8: Medeldifferens mellan modellerna. Illustration av resultaten i Tabell 1.

5. Slutsatser

Den tidsdiskreta modellen som presenteras i artikeln är, enligt vår kännedom, den första av sitt slag som utvecklats för en s.k. ”flexible jobshop”. Denna modell överträffar vida ingenjörsmodellen vad gäller erforderlig beräkningstid.

Längden ℓ av den diskreta tidsperioden är en viktig parameter eftersom den direkt påverkar problemets storlek, och därmed både den mängd datorminne samt den beräkningstid som krävs för att ta fram optimala scheman. Den tidsdiskreta modellen (1) med $\ell = 1h$ finner ett optimalt schema för verkliga scenarier från multitaskcellen på mindre än en minut. Dessutom är skillnaden i resultat mellan denna modell och modeller med lägre värden på ℓ mycket liten. Den tidsdiskreta modellen har således goda förutsättningar att fungera praktiskt, eftersom den både är snabb och ger goda resultat.

Referenser

- van den Akker, J., Hurkens, C., and Savelsberg, M. (2000) *Time-indexed formulations for machine scheduling problems: Column generation*, *INFORMS Journal on Computing*, Vol. 12, pp. 111–124.
- Baykasoglu, A. and Özbakir, L. (2010) *Analyzing the effect of dispatching rules in the scheduling performance through grammar based flexible scheduling system*, *International Journal of Production Economics*, Vol 124, pp. 269–381.
- Brucker, P. (2007) *Scheduling Algorithms*, 5th ed., Section 6.4, Springer-Verlag, Berlin Heidelberg New York.
- Brucker, P., Jurisch, B., and Krämer, A. (1997) *Complexity of scheduling problems with multi-purpose machines*, *Annals of operations research*, Vol 70, pp. 57-73.
- Jain, A. and Meeran, S. (1999) *Deterministic job-shop scheduling: Past, present and future*, *European Journal of Operational Research*, Vol 113, pp. 390–434.
- Manne, A.S. (1960) *On the job-shop scheduling problem*, *Operations Research*, Vol 8, pp. 219–223.
- Taha, H.A., (2007) *Operations Research: An Introduction*, 8th ed., Pearson Education, Inc., Upper Saddle River, New Jersey, USA.
- Thörnblad, K. och Kjellsdotter Ivert, L. (2010) *A comparison of schedules resulting from priority rules and mathematical optimization for a real production cell*, *PLANs forsknings- och tillämpningskonferens 2010, Skövde*.

Thörnblad, K., Strömberg, A.-B., Almgren, T., and Patriksson, M. (2010) *Optimization of schedules for a multitask production cell, 22nd Nofoma conference proceedings, Kolding, Denmark, 2010.*

Wolsey, L. (1997) *MIP modelling of changeovers in production planning and scheduling problems, European Journal of Operational Research, Vol. 99, pp. 154–165.*