

Programação da produção de gabaritos de montagem com restrições de adjacência na indústria aeronáutica

Bruno Jensen Virginio da Silva^{a*}, Reinaldo Morabito^b,
Horacio Hideki Yanasse^c

^{a*}brunojen@gmail.com, UFSCar, Brasil

^bmorabito@ufscar.br, UFSCar, Brasil

^choracio@lac.inpe.br, INPE, Brasil

Resumo

Esse artigo trata de um problema de programação da produção característico da indústria aeronáutica, envolvendo estruturas especiais de montagem chamadas gabaritos, compostas de diversos postos de trabalho em paralelo, na montagem de partes das aeronaves. Tarefas devem ser programadas para serem executadas nestes postos de trabalho de maneira a minimizar o *makespan*; porém, além das restrições usuais, como prazos de entrega das tarefas e precedências entre as tarefas, existem também restrições que impedem que duas tarefas possam ser executadas ao mesmo tempo em dois postos de trabalho adjacentes no gabarito. Com base no estudo de casos práticos de programação de gabaritos de montagem de uma empresa aeronáutica, propõe-se um modelo de programação linear inteira mista. As soluções geradas pelo modelo foram implementadas na prática com ganhos tanto na utilização dos gabaritos de montagem estudados quanto na utilização da mão de obra envolvida.

Palavras-chave

Programação da produção. Gabaritos de montagem. Indústria aeronáutica. Programação linear inteira mista.

1. Introdução

No Brasil, a indústria aeronáutica vem crescendo de maneira significativa nas últimas décadas. Nos últimos oito anos, a principal empresa deste segmento esteve entre as maiores exportadoras brasileiras. Em 2008, por exemplo, esta empresa foi responsável por cerca de 2% de todas as exportações, o que representa cerca de US\$ 5 bilhões (BRASIL, 2009). A tendência continua a ser de participação crescente, com a entrada da empresa em novos mercados de aviação (EMBRAER, 2008, 2009) e o lançamento de novos produtos, e isso aumenta ainda mais a sua importância para a economia brasileira. Neste contexto, oportunidades para melhorar a competitividade da indústria aeronáutica brasileira são particularmente relevantes, principalmente com o ingresso de novos concorrentes diretos no mercado de aviação comercial (OTTOBONI, 2007; SUKHOI, 2007; OLMOS, 2008; BOWKER, 2009).

Pode-se dizer de maneira simplificada que a produção de aviões segue basicamente duas etapas: fabricação e montagem. A montagem pode ser vista sob vários aspectos, desde a montagem de pequenos subconjuntos até a montagem estrutural dos aviões propriamente. Nas operações de montagem são comumente utilizados os gabaritos de montagem. A indústria aeronáutica utiliza essa técnica para garantir as condições de projeto dos produtos montados e a sua compatibilidade com peças de montagens posteriores. Gabaritos variam em tamanho e complexidade, dependendo das peças envolvidas e das operações nele realizadas. Geralmente, as operações de montagem realizadas nos gabaritos são manuais, sendo os operadores os responsáveis pelo posicionamento e montagem dos componentes e subconjuntos. Assim, um problema relevante na indústria aeronáutica é a programação da produção nos gabaritos de montagem.

Como os processos envolvidos em geral são manuais, grande quantidade dos operadores que trabalham na produção de aviões se dedica às operações de montagem em gabaritos, e o estudo da programação dessas operações é importante para a adequação da utilização da mão de obra, que é qualificada, especializada, intensiva e cara. Além disso, os gabaritos são específicos dos produtos neles montados; logo, a quantidade de aviões a ser produzida depende da disponibilidade e capacidade de utilização dos gabaritos de montagem. Em alguns gabaritos, devido às suas especificidades, apenas um tipo de produto pode ser montado. Esse é o caso, por exemplo, dos gabaritos de montagem das asas de aviões de médio porte, que são partes grandes, complexas e específicas do avião. Estas partes necessitam de um gabarito de montagem para cada asa, direita e esquerda. Em outros gabaritos podem ser montados produtos diferentes, um de cada vez. É o caso, por exemplo, de alguns segmentos da fuselagem do avião, que podem ser montados num mesmo gabarito depois de configurá-lo adequadamente para cada conjunto a ser montado. Existem ainda gabaritos que podem montar vários produtos diferentes ao mesmo tempo. É o caso dos grandes subconjuntos, que são compostos por várias partes segmentadas, ou de algumas partes da fuselagem muito similares, mas com tamanhos variados para aviões diferentes.

No presente trabalho, é estudado esse último tipo de gabarito, com a particularidade de que quando alguns produtos são montados usando alguns postos de trabalho do gabarito, outros produtos não podem ser montados ao mesmo tempo em postos de trabalho adjacentes. Essa limitação é aqui denominada de restrição de adjacência. A literatura que aborda problemas de programação de tarefas em máquinas é extensa (CONWAY; MAXWELL; MILLER, 1967; BAKER, 1974; COFFMAN, 1976; HAX; CANDEA, 1984; MORTON; PENTICO, 1993; NAHMIA, 2001; PINEDO, 2005, 2008; POTTS; STRUSEVICH, 2009, só para citar alguns). No entanto, os autores encontraram poucos estudos publicados que tratam de programação da produção com restrições de adjacência similares às consideradas neste trabalho. Restrições de adjacência em outros contextos de planejamento e programação da produção foram consideradas, por exemplo, em Murray (1999) e Weintraub e Murray (2006), no planejamento de colheitas de madeira em florestas de manejo evitando-se colheitas em áreas adjacentes no mesmo período; em Duin e Van der Sluis (2006), na programação de recursos adjacentes no problema de alocação de atendentes em balcões de aeroportos; e em Santos et al. (2007, 2008), no planejamento de rotações de culturas também com restrições de adjacência. As abordagens utilizadas nesses trabalhos,

no entanto, não podem ser diretamente aplicadas ao problema focado neste estudo.

Neste trabalho considera-se o problema de programação da produção de gabaritos de montagem com restrições de adjacência encontrado no processo de montagem de jatos comerciais e executivos de uma empresa aeronáutica. Apresenta-se um modelo de programação linear inteira mista para representar o problema, que foi implementado computacionalmente na linguagem de modelagem GAMS e resolvido utilizando o *software* de otimização CPLEX. Trata-se, portanto, de uma pesquisa baseada em modelagem quantitativa com abordagem de pesquisa empírico-normativa para experimentos (BERTRAND; FRANSOO, 2002; MORABITO; PUREZA, 2009), que reporta os resultados a partir da aplicação em um caso real.

Este artigo está organizado em seis seções. Na seção 2, o problema de programação da produção de gabaritos de montagem é descrito em detalhes, contextualizando o ambiente de produção e aspectos do seu funcionamento. Na seção 3, o modelo matemático utilizado para representar o problema é proposto. Na seção 4 são apresentados e analisados os resultados computacionais do modelo matemático, utilizando o aplicativo GAMS/CPLEX. Na seção 5 é descrita a implantação da solução no chão de fábrica da empresa estudada e, finalmente, na seção 6 são discutidas as considerações finais deste estudo e algumas perspectivas para pesquisa futura.

2. Descrição do problema

Como mencionado anteriormente, a produção de aviões pode ser dividida, de maneira simplificada, em fabricação e montagem – o problema estudado relaciona-se à montagem. Problemas de montagem podem ser separados em três tipos: montagem de subconjuntos, montagem estrutural e montagem final. Montagem de subconjuntos é qualquer montagem que envolva duas ou mais peças. Montagens que envolvem poucas peças e peças pequenas costumam utilizar apenas ferramentas manuais e, em geral, não precisam de gabaritos. Os chamados grandes subconjuntos possuem uma complexidade de processo de montagem bem maior e geralmente envolvem várias peças, utilizando gabaritos de montagem, são de tamanho maior, variando entre cerca de um e três metros, e são fornecidos diretamente para uma etapa de montagem estrutural ou final. Esse é o caso do problema aqui considerado.

A montagem estrutural é aquela que monta as partes estruturais do avião – fuselagem, asas e empenagem. A montagem final é a etapa em que todas as partes são montadas, ou seja, as partes da

fuselagem entre si e com outras partes, como dianteira, traseira, estabilizador horizontal e vertical, asas, além da equipagem elétrica e eletrônica; enfim, todas as partes do avião são incorporadas, é feito o vôo de produção e a entrega ao cliente. Detalhes sobre as estruturas dos aviões, sua complexidade e algumas técnicas de produção podem ser encontradas em Niu (1988), Megson (1999) e Howe (2004).

A programação de produção dos gabaritos é feita de diferentes maneiras, com diferentes graus de detalhamento, em função da importância da montagem a que se destina. No caso focado, a programação da produção é feita de modo pouco detalhado. A área responsável pelo planejamento e controle da produção apenas fixa datas de entrega das peças. O detalhamento da programação é responsabilidade da área de produção, sendo ela a responsável por iniciar as montagens, distribuir as tarefas pelas pessoas da operação e administrar o avanço da montagem. As quantidades de subconjuntos a serem produzidos são conhecidas porque são baseadas na cadência de produção dos aviões. Estas cadências nada mais são do que a quantidade mensal de aviões de certo modelo que será produzida. Elas são conhecidas antecipadamente, e variam pouco com o tempo, já que pequenas mudanças na cadência de produção podem significar grandes mudanças na dinâmica da produção.

A ordem de grandeza das cadências varia de algumas unidades para algumas dezenas. As maiores cadências praticadas na empresa estudada foram da ordem de poucas dezenas de aviões por mês. Alguns modelos de aeronaves produzidas, que são os jatos comerciais, utilizam dois subconjuntos de peças por avião do tipo estudado neste trabalho. Assim, se a cadência de produção desses modelos for de 10 aviões por mês, a demanda dos subconjuntos será então de 20 unidades por mês.

Na Figura 1 ilustra-se um exemplo de um subconjunto composto de oito peças (denominadas peças 1, 2,..., 8), que após montadas e encaixadas uma na outra formam o subconjunto. Cada peça possui apenas um posto de trabalho específico no gabarito, apresentado na Figura 2, para ser montada. A peça 1 do subconjunto deve ser montada no posto de trabalho 1 do gabarito, a peça 2 deve ser montada no posto de trabalho 2, e assim sucessivamente, para todas as 8 peças do subconjunto.

As montagens das oito peças do subconjunto são realizadas individualmente. Cada peça é formada por um conjunto de componentes. A operação de montagem realizada nos postos de trabalho do gabarito consiste na montagem desses componentes que compõem cada peça do subconjunto. As oito

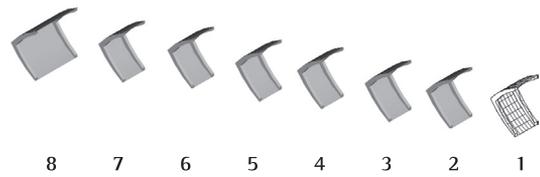


Figura 1. As oito peças que compõem o subconjunto.

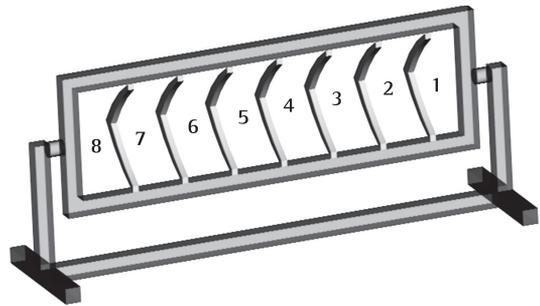


Figura 2. Gabarito de montagem do subconjunto da Figura 1, com oito postos de trabalho.

peças do subconjunto são montadas individualmente no gabarito e são, posteriormente, encaixadas para formar este subconjunto (i.e., uma parte do avião) apenas em outra etapa de montagem na linha de produção, que não é considerada nesse trabalho. Isso significa que cada peça do subconjunto pode ser montada e entregue individualmente, mesmo fazendo parte do mesmo subconjunto do mesmo avião. A administração das peças consiste basicamente na administração dos prazos de entrega de cada peça, que estão relacionados a cada avião em processo de montagem.

A restrição de adjacência surge porque não há espaço físico para que os operadores montem ao mesmo tempo peças do subconjunto que utilizam postos de trabalho (estações do gabarito) adjacentes. Por exemplo, se um operador está montando a peça 5 do subconjunto no posto de trabalho 5 (Figuras 1 e 2), os postos de trabalho 4 e 6 ficam bloqueados fisicamente e não podem ser utilizados. Neste caso, os postos de trabalho 1, 2, 3, 7 e 8 ficam disponíveis para montar as peças do subconjunto que neles devem ser montadas.

A montagem de cada peça acontece, em geral, em duas etapas. A primeira etapa consiste na montagem no gabarito e, em seguida, uma complementação da montagem que é realizada fora do gabarito, nas chamadas *bancadas*. Nos gabaritos são feitos os posicionamentos dos componentes que formam a parte do subconjunto, utilizando furos e batentes que o gabarito possui como referência para posicionamento

desses componentes. Depois disso, as peças vão para as bancadas, para os processos de furação e rebiteagem, que não precisam do gabarito para serem executados, já que o posicionamento dos componentes já foi feito. A segunda etapa da montagem se inicia novamente no gabarito, terminando com uma complementação realizada nas bancadas. Isso ocorre por restrições técnicas, pois nem todos os componentes podem ser totalmente referenciados na primeira etapa do gabarito, sendo necessária uma segunda etapa para a maioria das peças. Em alguns casos pode-se ter uma terceira etapa de montagem da peça, também composta de uma operação no gabarito, seguida de uma operação nas bancadas.

Considera-se cada par de operações, uma no gabarito e outra nas bancadas, uma tarefa. Na Figura 3 representa-se uma rede de precedência das montagens de cada peça – cada uma composta de apenas duas tarefas. Em branco estão as operações executadas no gabarito e em cinza, as operações executadas nas bancadas. Os números dentro da atividade (1 a 16) identificam as tarefas associadas à montagem das peças do subconjunto e às operações de montagem no gabarito (1) ou nas bancadas (2).

Nesta rede da Figura 3 cada peça do subconjunto necessita de duas tarefas para ser montada e cada tarefa possui duas operações. Por exemplo, a peça 1 do subconjunto necessita das tarefas 1 e 2 para ser montada. As tarefas 1 e 2 são compostas das operações 1 e 2 (i.e., 1.1 e 1.2, e 2.1 e 2.2, respectivamente).

O gráfico de Gantt do programa de produção de um subconjunto (uma parte do avião) tem um padrão bastante singular. Na Figura 4 é apresentado um exemplo de programação da produção considerando apenas as primeiras três peças (peças 1, 2 e 3) e suas respectivas tarefas (tarefas 1, 2,..., 6) nas três primeiras posições do gabarito específicas para montagem destas peças. As operações em branco são as montagens no gabarito e as operações em cinza são as montagens nas bancadas. O gráfico apresenta diversos espaços vazios que indicam postos de trabalho sem montagens, ociosos ou bloqueados devido a montagens em postos de trabalho adjacentes.

Observe que postos de trabalho adjacentes, ou seja, postos de trabalho com números sucessivos, nunca apresentam operações brancas sendo executadas ao mesmo tempo, devido às restrições de adjacência no gabarito. Algumas tarefas referentes a uma mesma parte do subconjunto possuem restrição de precedência. Por exemplo, a tarefa 2 só pode ser iniciada depois de terminada a tarefa 1. O mesmo acontece com as tarefas 4 e 3, e as tarefas 6 e 5. A programação apresentada na Figura 4 respeita essas restrições.

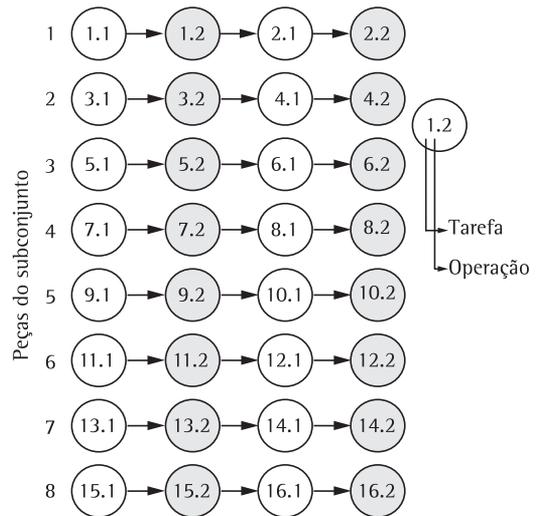


Figura 3. Rede de precedência da montagem do subconjunto.

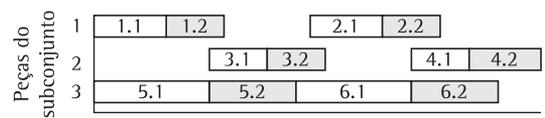


Figura 4. Gráfico de Gantt de uma programação de três peças do subconjunto para o gabarito e bancadas.

3. Modelagem matemática

O problema considerado é de programação em máquinas paralelas distintas com restrições de adjacência entre tarefas em execução dependentes das máquinas. O problema é basicamente programar a montagem das peças de um subconjunto no gabarito, com o objetivo de minimizar o tempo total para completar as tarefas (*makespan*). Além das restrições de adjacência, têm-se datas de disponibilidade e prazos de entrega das tarefas, restrições de precedência entre as tarefas de montagem e restrições de tecnologia (i.e., o posto de trabalho do gabarito em que cada tarefa deve ser executada).

A solução deste problema de programação da produção não nos parece ser imediata ou trivial. Muitos problemas de programação da produção são NP-difíceis (LAWLER et al., 1993; LEUNG, 2004) e este parece também ser o caso deste problema focado. Mostrar a sua complexidade está além do escopo deste trabalho e é um tópico para pesquisa futura.

A seguir é apresentado um modelo matemático linear inteiro misto para representar o problema. As variáveis de decisão utilizadas consideram a ordem relativa em que as tarefas são executadas, o instante de início de cada tarefa e, o instante final de execução de todas as tarefas.

Sejam os seguintes parâmetros do modelo:

- p_j : duração da operação no gabarito da tarefa j ($j = 1, 2, \dots, J$) em que J é o número total de tarefas;
- q_j : duração da operação nas bancadas da tarefa j ;
- r_j : instante de tempo que a tarefa j fica disponível para programação;
- d_j : prazo de entrega da tarefa j .

Além desses parâmetros, definem-se os seguintes conjuntos de pares de tarefas (j, k) , $j \neq k$:

- $W = \{(j, k) \mid \text{as tarefas } j \text{ e } k \text{ são executadas no mesmo posto de trabalho do gabarito}\}$
- $A = \{(j, k) \mid \text{a execução da tarefa } j \text{ precede a execução da tarefa } k\}$
- $B = \{(j, k) \mid \text{as tarefas } j \text{ e } k \text{ são montadas em postos de trabalho adjacentes no gabarito}\}$

As variáveis de decisão do modelo são:

- $y_{jk} = \begin{cases} 1 & \text{se a tarefa } j \text{ é programada antes da tarefa } k \\ 0 & \text{caso contrário} \end{cases}$
- t_j = instante de tempo do início da tarefa j
- t_f = o instante de término de execução de todas as tarefas, denominado *makespan*

O problema é modelado então como se segue:

$$\text{Min } t_f \quad (1)$$

$$t_j + p_j + q_j \leq t_f \quad j = 1, \dots, J \quad (2)$$

$$t_j \geq r_j \quad j = 1, \dots, J \quad (3)$$

$$t_j + p_j + q_j \leq d_j \quad j = 1, \dots, J \quad (4)$$

$$t_k + M \cdot (1 - y_{jk}) \geq t_j + p_j \quad \text{para todo par } (j, k) \in W \quad (5)$$

$$t_j + M \cdot y_{jk} \geq t_k + p_k \quad \text{para todo par } (j, k) \in W \quad (6)$$

$$t_k \geq t_j + p_j + q_j \quad \text{para todo par } (j, k) \in A \quad (7)$$

$$t_k + M \cdot (1 - y_{jk}) \geq t_j + p_j \quad \text{para todo par } (j, k) \in B \quad (8)$$

$$t_j + M \cdot y_{jk} \geq t_k + p_k \quad \text{para todo par } (j, k) \in B \quad (9)$$

$$\begin{aligned} t_j &\geq 0 \quad j = 1, \dots, J, \\ y_{jk} &\in \{0, 1\}, \quad j = 1, \dots, J, \\ k &= 1, \dots, J, \quad j \neq k \end{aligned} \quad (10)$$

A função objetivo (1) junto com as restrições (2) garantem que o *makespan* t seja mínimo e igual ao instante de término da última tarefa programada. As restrições (3) garantem que cada tarefa j só possa ser programada após seu instante de disponibilidade r_j para produção. As restrições (4) garantem que cada tarefa j seja completada antes do seu prazo de entrega d_j . As restrições (5) e (6) garantem que duas tarefas j e k processadas no mesmo posto de trabalho do gabarito não tenham suas operações no gabarito programadas ao mesmo tempo. Note que essas restrições são definidas somente para os pares de tarefas j e k montadas no mesmo posto de trabalho. Note também que as restrições (5) e (6) são disjuntivas, ou seja, quando uma estiver ativa, a outra se torna redundante, e vice-versa (ARENALES et al., 2006). O parâmetro M é um número positivo suficientemente grande, por exemplo, definido como:

$$\sum_{j=1}^J (p_j + q_j) \quad (11)$$

As restrições (7) garantem a relação de precedência de execução entre duas tarefas j e k . Note que elas são definidas apenas para os pares de tarefas j e k em que a execução da tarefa j deve preceder a execução da tarefa k . As restrições (8) e (9) garantem que duas tarefas j e k que utilizam postos de trabalho adjacentes no gabarito não tenham suas execuções no gabarito programadas ao mesmo tempo. Note que essas restrições também são disjuntivas e são definidas somente para os pares de tarefas j e k montadas em postos de trabalho adjacentes. Finalmente, as restrições (10) referem-se ao domínio das variáveis.

4. Resultados computacionais

Nessa seção, são apresentados, inicialmente, os resultados da aplicação do modelo (1)-(10) em um experimento ilustrativo (exemplo 1) com dados fictícios. Em seguida, são apresentados os resultados de outros dois experimentos (exemplos 2 e 3) utilizando dados reais da empresa estudada. A diferença entre os exemplos 2 e 3 é que a programação foi feita em momentos diferentes da empresa, envolvendo mudanças no processo de montagem, conseqüentemente, na duração das tarefas e no número de etapas. O modelo (1)-(10) foi implementado no *software* GAMS 19.3 (*Generic Algebraic Modeling System*) e resolvido pelo *solver* CPLEX 7.0 em um computador com 512 MB RAM e processador de 1.8 GHz.

4.1. Exemplo ilustrativo

O exemplo ilustrativo é utilizado para explicar as especificidades do problema de programação da

produção tratado. Considere um gabarito fictício com apenas três postos de trabalho, 1, 2 e 3, e que a produção de seis tarefas ($j = 1, \dots, 6$) precisa ser programada neste gabarito. As durações (em unidade de tempo, u.t.) das operações e os respectivos postos de trabalho das tarefas são mostrados na Tabela 1.

Nesse caso, o subconjunto ilustrativo é formado por três peças, em que cada peça possui duas tarefas, cada uma com duas operações, ou seja, um total de seis operações. As tarefas 1 e 2 só podem ser executadas no posto de trabalho 1, as tarefas 3 e 4 só podem ser executadas no posto de trabalho 2, e as tarefas 5 e 6 só podem ser executadas no posto de trabalho 3. Considera-se que todas as tarefas estão disponíveis no instante zero (i.e., $r_j = 0, j = 1, 2, \dots, 6$), e que o instante 50 é o prazo de entrega de todas as tarefas (i.e., $d_j = 50, j = 1, 2, \dots, 6$).

O conjunto W , que contém os pares de tarefas que são processadas no mesmo posto de trabalho, é dado por: $W = \{(1, 2), (3, 4), (5, 6)\}$. Da Tabela 1 observa-se que as tarefas 1, 3 e 5 têm precedência em relação às tarefas 2, 4 e 6, respectivamente. Logo, o conjunto A , que representa as relações de precedência das tarefas, é dado por: $A = \{(1, 2), (3, 4), (5, 6)\}$. Neste exemplo, o posto de trabalho 1 é adjacente ao 2, o posto de trabalho 2 é adjacente aos postos de trabalho 1 e 3, e o posto de trabalho 3 é adjacente ao posto de trabalho 2. Portanto, o conjunto B , que contém os pares de tarefas que são montadas em postos de trabalhos adjacentes, é dado por: $B = \{(1, 3), (1, 4), (2, 3), (2, 4), (3, 5), (3, 6), (4, 5), (4, 6)\}$.

Na Figura 5 apresenta-se o gráfico de Gantt gerado a partir da solução do modelo para o exemplo 1. O tempo de resolução utilizando o GAMS/CPLEX foi de menos de 1 segundo. Os números que indicam as tarefas e os gráficos das operações destas tarefas no gabarito (p_j) e nas bancadas (q_j) estão apresentados separadamente. Perceba que o gráfico de Gantt possui espaços em branco, mostrando que, em alguns intervalos de tempo, os postos de trabalho do gabarito e as bancadas ficam sem processar tarefas, por estarem bloqueadas por tarefas sendo processadas em postos de trabalho adjacentes ou aguardando o término de tarefas precedentes.

Também existem intervalos de tempo com folgas; por exemplo, a operação da tarefa 3 executada nas bancadas, podendo ser iniciada dentro de um intervalo de tempo que não compromete a solução ótima. Note que, apesar de todas as tarefas estarem disponíveis para serem iniciadas no instante zero, nesta solução a tarefa 3 precisa esperar para ser iniciada. Além disso, a tarefa 4 também fica aguardando o seu processamento por um período. O tempo total para completar as tarefas, *makespan*, é igual a 41.

Tabela 1. Duração das operações das tarefas a serem executadas no gabarito fictício.

Peça do subconjunto	Tarefa (j)	Posto de trabalho no gabarito	Tempo no gabarito (p_j)	Tempo nas bancadas (q_j)	Tarefas precedentes
1	1	1	5	10	-
	2	1	5	10	1
2	3	2	4	9	-
	4	2	4	9	3
3	5	3	8	12	-
	6	3	8	12	5

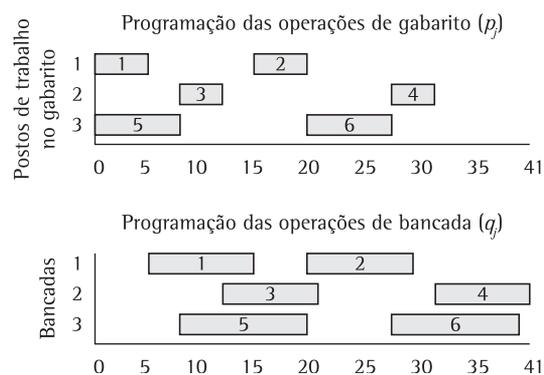


Figura 5. Gráfico de Gantt com a solução para o exemplo ilustrativo.

Como o prazo de entrega de todas as tarefas é 50, todas elas terminam dentro do prazo.

Embora este exemplo ilustrativo seja relativamente simples, sua solução já não é trivial. A dificuldade de determinar uma boa programação viável cresce com o aumento do número de tarefas e postos de trabalho do gabarito.

4.2. Exemplos reais

O primeiro exemplo real, exemplo 2, refere-se a dados coletados na fase inicial de produção, que é ainda uma fase de aprendizado em que a maior cadência de produção ainda não foi atingida. A implicação disso é que os dados estão sujeitos a mudanças no futuro devido à evolução da curva de aprendizagem (SCHMENNEN, 1990).

Este exemplo possui oito peças que formam um subconjunto, duas peças são montadas em três etapas envolvendo tarefas com 6 operações intercaladas de montagem, três de gabarito e três de bancada, conforme apresentado na Tabela 2 (tarefas 13 a 24). Além disso, há restrições de precedência entre as tarefas, por exemplo, a tarefa 13 deve preceder a tarefa 14, que por sua vez deve preceder a tarefa 15.

Tabela 2. Duração (em u.t.) das operações das peças do subconjunto para o exemplo 2.

Subconjunto	Peça do subconjunto	Tarefa (j)	Posto de trabalho no gabarito	Tempo no gabarito (p_j)	Tempo nas bancadas (q_j)	Tarefas precedentes
1	1	1	1	60	75	-
		2	1	75	240	1
2	1	3	1	60	75	-
		4	1	75	240	3
1	2	5	2	90	160	-
		6	2	50	280	5
2	2	7	2	90	160	-
		8	2	50	280	7
1	3	9	3	90	140	-
		10	3	50	280	9
2	3	11	3	90	140	-
		12	3	50	280	11
1	4	13	4	90	150	-
		14	4	90	150	13
2	4	15	4	45	60	14
		16	4	90	150	-
2	4	17	4	90	150	16
		18	4	45	60	17
1	5	19	5	90	150	-
		20	5	90	150	19
2	5	21	5	45	60	20
		22	5	90	150	-
2	5	23	5	90	150	22
		24	5	45	60	23
1	6	25	6	90	180	-
		26	6	60	120	26
2	6	27	6	90	180	-
		28	6	60	180	28
1	7	29	7	90	180	-
		30	7	60	120	30
2	7	31	7	90	180	-
		32	7	60	120	32
1	8	33	8	90	200	-
		34	8	60	360	34
2	8	35	8	90	200	-
		36	8	60	360	36

Também a tarefa 16 deve preceder a tarefa 17, que por sua vez deve preceder a tarefa 18.

O problema é composto por 18 tarefas em que, geralmente, as operações realizadas nas bancadas têm duração maior que as operações realizadas nos gabaritos. Para proteger interesses da empresa, a unidade de tempo (u.t.) das durações das operações não é revelada e os valores originais foram multiplicados por um coeficiente. A programação de montagem de dois subconjuntos deve ser realizada (equivalente à quantidade de subconjuntos que cada produto final precisa – veja detalhes na seção 5), portanto, tem-se uma duplicação de tarefas totalizando 36.

Na Tabela 2 são apresentados os dados das 36 tarefas. Perceba que existem tarefas similares,

decorrentes de peças similares, mas referentes a cada um dos dois subconjuntos. Os conjuntos W , A e B para este exemplo 2 estão detalhados no Anexo 1.

O modelo correspondente a este exemplo 2 possui 464 variáveis contínuas, 426 variáveis binárias e 987 equações, e o GAMS/CPLEX demorou cerca de 5 segundos para encontrar a solução. Na Figura 6 apresentam-se os gráficos de Gantt gerados a partir da solução obtida neste problema. O primeiro gráfico é o programa das operações do gabarito e o segundo gráfico, o programa das operações das bancadas. O tempo para completar as operações do gabarito foi de 915 u.t., e o das bancadas foi de 975 u.t.

Como pode ser observado no primeiro gráfico da Figura 6, a solução obtida obedece as restrições

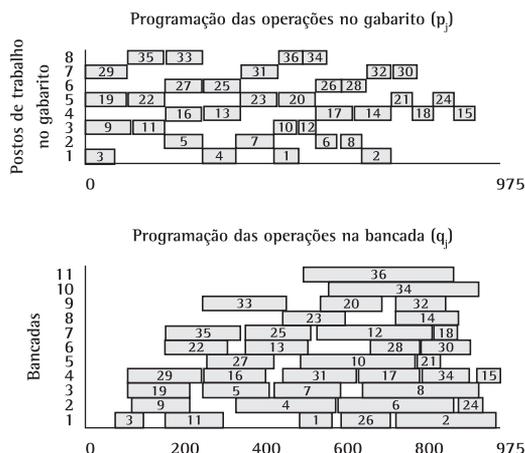


Figura 6. Gráfico de Gantt da programação das operações de gabarito e de bancadas do exemplo 2.

de adjacência e de precedência entre tarefas. Na programação das bancadas (segundo gráfico de Gantt da Figura 6), as operações foram simplesmente agrupadas em cada linha do gráfico, sem preocupação em otimizar o número de pessoas, porém respeitando os tempos de início das operações das bancadas. Observa-se que no início da programação, a quantidade de tarefas em processamento nas bancadas é menor por conta de as operações e/ou tarefas que eventualmente as precedem estarem sendo executadas ou precisarem ser executadas no gabarito. O inverso acontece no final da programação, em que há uma quantidade maior de tarefas em processamento nas bancadas e um número menor de operações no gabarito.

O segundo exemplo real, exemplo 3, difere do exemplo 2 nos seguintes aspectos: no exemplo 3 todas as tarefas possuem no máximo 4 operações de montagem (no gabarito e nas bancadas) e a duração de algumas operações é diferente, devido a diminuições da duração ou transferência de partes do processo entre as operações. Estas são decorrentes de evoluções técnicas do processo de montagem efetuadas pela engenharia de processos e que mudam os dados e a solução do problema.

Na Tabela 3 são apresentados os dados do exemplo 3. Nesse caso, são necessárias 30 tarefas para montar as peças de dois subconjuntos inteiros, seis tarefas a menos que no modelo 2. A peça 3 do subconjunto possui apenas uma tarefa (tarefas 9 e 10, uma para cada um dos dois subconjuntos). Os respectivos conjuntos *W*, *A* e *B* para este exemplo estão detalhados no Anexo 1.

O modelo para o exemplo 3 possui 310 variáveis contínuas, 278 variáveis binárias e 667 equações, e o CPLEX demorou menos de 1 segundo para encontrar a solução. O tempo total para completar as tarefas, o

makespan, foi de 815 u.t. As operações do gabarito terminam no tempo 615 u.t. Comparando-se os diagramas das Figuras 6 e 7, percebe-se que embora os problemas se refiram à montagem de um mesmo subconjunto de peças, a mudança no número de tarefas e na duração delas (evoluções técnicas do processo, aprendizado) gera programas bem diferentes.

A diminuição do *makespan* no exemplo 3, em comparação com o 2, se deve à redução do número de tarefas e na duração total da montagem das peças do subconjunto verificada para quase todas elas, exceto as peças 2 e 8.

Cada montagem realizada nas bancadas necessita de uma pessoa. Portanto, analisando-se o diagrama de Gantt das operações nas bancadas, pode-se contabilizar o número de pessoas necessárias nas bancadas a cada instante para efetuar a montagem (verificando o número de operações sendo realizadas em paralelo). Note que o número máximo de pessoas utilizado para as operações de bancadas na solução obtida para os exemplos 2 e 3 foi 11.

5. Implementação do novo procedimento para definição da programação das tarefas no chão de fábrica

Na prática, a programação da produção no chão de fábrica onde é realizada a montagem descrita no problema era feita de modo manual e utilizando o bom senso. Prazos de entrega para os subconjuntos eram estabelecidos sugeridos pelo sistema MRP (*Materials Requirement Planning*) da empresa e a programação era definida utilizando como política a montagem, primeiro, das peças dos subconjuntos com prazo de entrega mais curto ou mais atrasadas (EDD – *Earliest Due Date*). A programação das peças a serem montadas era feita para no máximo um dia. Esse modo de organizar a montagem é referido neste trabalho como programação por peças.

Outra maneira de organizar a montagem seria a programação por avião. Estabelecida a cadência de produção dos aviões, pode-se calcular o tempo de ciclo para atender à demanda desta cadência de produção. Por exemplo, para uma cadência de 10 aviões/mês e mês com 20 dias úteis de produção, o tempo de ciclo é:

$$t_{ciclo} = \frac{20 \text{ dias}}{10 \text{ aviões/mês}} = 2 \text{ dias} \tag{12}$$

Assim, cada par de subconjuntos dos 10 aviões deve ser montado a cada dois dias úteis do mês.

Na Figura 8 ilustram-se os programas de produção de cada uma das duas formas de programar a produção

Tabela 3. Duração (em u.t.) das peças do subconjunto para o exemplo 3.

Subconjunto	Peça do subconjunto	Tarefa (j)	Posto de trabalho no gabarito	Tempo no gabarito (p_j)	Tempo nas bancadas (q_j)	Tarefas precedentes
1	1	1	1	25	35	-
		2	1	20	235	1
2	1	3	1	25	35	-
		4	1	20	235	3
1	2	5	2	75	145	-
		6	2	45	355	5
2	2	7	2	75	145	-
		8	2	45	355	7
1	3	9	3	55	200	-
		10	3	55	200	-
1	4	11	4	60	200	-
		12	4	30	200	11
2	4	13	4	60	200	-
		14	4	30	200	13
1	5	15	5	40	150	-
		16	5	30	200	15
2	5	17	5	40	150	-
		18	5	30	200	17
1	6	19	6	60	85	-
		20	6	30	200	19
2	6	21	6	60	85	-
		22	6	30	200	21
1	7	23	7	35	60	-
		24	7	35	150	23
2	7	25	7	35	60	-
		26	7	35	150	25
1	8	27	6	50	305	-
		28	6	60	330	27
2	8	29	7	50	305	-
		30	7	60	330	29

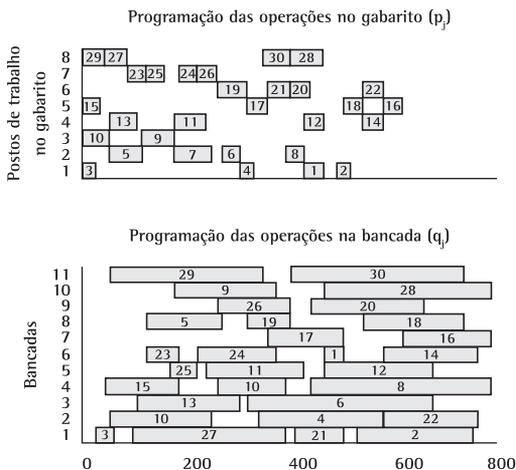


Figura 7. Gráfico de Gantt da programação das operações de gabarito e de bancadas do exemplo 3.

citadas anteriormente: programação por peças e programação por avião. No primeiro programa, como a montagem é feita peça por peça, considera-se

que a tarefa de montagem do avião inicia quando começa a montagem da primeira de qualquer uma das peças do par de subconjuntos deste avião e termina quando a última peça está pronta. Assim, o tempo de montagem de cada avião pode variar. Note que o avião 1 começa a produção no início do período e o avião 2 começa a montagem no tempo 2 e ambas as montagens terminam no tempo 5 u.t. As peças dos aviões 5 e 6, que são montadas pelas mesmas operações, demoram 7 e 6 u.t., respectivamente.

Por outro lado, na programação por avião da Figura 8, apenas as peças de um único avião são montadas de cada vez. As peças de um segundo avião são montadas apenas quando todas as peças do primeiro avião já estão montadas, e assim por diante. Assim, o tempo de montagem de cada avião não varia e é igual ao seu tempo de ciclo.

Comparando-se as duas formas de programação, na programação por peças os tempos totais de montagem das peças podem variar de avião para avião, mesmo que todos os aviões sejam iguais. Portanto, existe uma maior incerteza no tempo total

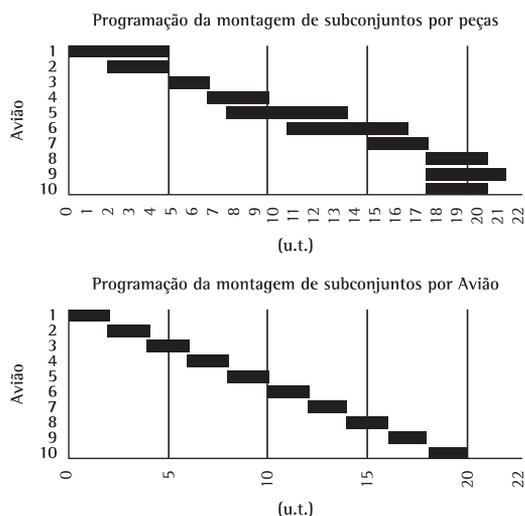


Figura 8. Programação por peça e programação por avião.

de duração da montagem das peças de um avião utilizando a programação por peças, comparado com a programação por avião. Esta incerteza é consequência da existência de maiores tempos de espera e estoque em processos do que na programação por avião, o que aumenta o *lead time* de produção. Em alguns casos, a programação por peças pode ser utilizada para melhorar o nivelamento dos recursos de produção, diminuindo a ociosidade dos recursos (HEIKE et al., 2001).

Para utilizar a programação por avião, foi necessário encontrar uma programação de produção viável para montar o avião dentro do tempo de ciclo dado por uma cadência. O modelo matemático apresentado nesse trabalho auxilia a empresa a realizar tanto a programação por avião quanto a programação por peças. Soluções como as dos exemplos 2 e 3, apresentados na seção 4 e que utilizam a programação por avião, podem ser então implantadas na produção, desde que o *makespan* resultante da solução seja menor ou igual ao tempo de ciclo. Observe que o modelo pode ser executado para diversas cadências na programação por avião, oferecendo subsídios ao administrador sobre a cadência máxima de produção de aviões com os gabaritos existentes e os requisitos de mão de obra.

Outras vantagens da programação por avião em relação à programação por peças são: menor variabilidade na solicitação de fornecimento de componentes e na entrega de subconjuntos prontos, menor *lead time* de produção, menor estoque em processo, menor quantidade de horas extras, melhor utilização da mão de obra, e um potencial aumento gradativo da produtividade como consequência da repetição do mesmo programa.

A mudança da programação utilizada pela empresa para a programação obtida pela solução do modelo resultou em uma diminuição de cerca de 15% das necessidades de mão de obra na montagem, com melhoria no atendimento dos prazos de entrega das peças, diminuindo os atrasos para próximos de zero. A implantação da programação por avião obtida pelo modelo contou com o apoio das lideranças do chão de fábrica, tirando dúvidas dos operadores desde o início da utilização do programa, administrando situações não previstas no programa e coletando sugestões que foram sendo implementadas em novos programas.

6. Conclusões

Os autores desconhecem na literatura outros trabalhos que foquem o problema de programação da produção de gabaritos abordado nesse artigo, embora este tenha implicações práticas importantes na indústria aeronáutica, conforme apresentado. As soluções obtidas pelo modelo proposto mostram vantagens em relação às soluções empregadas pela empresa, tanto na utilização dos gabaritos de montagem estudados quanto na utilização da mão de obra. Os tempos computacionais requeridos para encontrar estas soluções foram aceitáveis, considerando-se as decisões envolvidas.

Nos casos de variações na cadência de produção dos aviões, o modelo proposto pode fornecer subsídios importantes ao tomador de decisão no planejamento de capacidade e recursos humanos nos gabaritos e nas bancadas. Note-se também que com este modelo é possível realizar experimentos para estimar a cadência máxima da unidade de produção na fábrica e suas necessidades de mão de obra. Estes experimentos podem ser particularmente úteis em casos, por exemplo, de aumento nas vendas.

Em situações em que as soluções do problema de programação envolvam necessidades de mão de obra excessivas nas bancadas, uma possível pesquisa futura seria estender o modelo para considerar uma restrição de limitação de mão de obra. Para isso, algumas alternativas poderiam ser exploradas como, por exemplo, a utilização de heurísticas para, a partir da solução obtida pelo modelo (1)-(10), realocar as tarefas nos gabaritos e nas bancadas em função das limitações de mão de obra violadas nas bancadas; a utilização de modelos de programação de projetos com limitações de recursos (mão de obra) para representar o problema restrito; ou mesmo outros modelos de programação que considerem como função objetivo minimizar o número de tarefas processadas simultaneamente (ou minimizar a mão de obra total requerida), ao mesmo tempo restritos a um dado *makespan* escolhido em função do tempo de ciclo da cadência de produção.

Acredita-se que, com pequenas modificações, o modelo proposto neste trabalho pode ser adaptado para representar a programação da produção de outros gabaritos de montagem desta e de outras empresas aeronáuticas. Outra perspectiva interessante de pesquisa futura seria a adaptação e aplicação das abordagens na programação da produção de gabaritos de outras indústrias que apresentem restrições de adjacência similares às presentes na indústria aeronáutica, mas que possuam restrições ou funções objetivo diferentes do problema apresentado nesse trabalho, devido a, por exemplo, diferentes tecnologias e modos de execução empregados nestes gabaritos.

Agradecimentos

Os autores agradecem aos revisores anônimos, pelos úteis comentários e sugestões, e à Sílvia Regina Morales pela colaboração com a pesquisa. Este estudo contou com apoio financeiro da FAPESP e do CNPq.

Referências

- ARENALES, M. et al. *Pesquisa operacional: modelagem e algoritmos*. Rio de Janeiro: Elsevier, 2006.
- BAKER, K. *Introduction to sequencing and scheduling*. New York: John Wiley, 1974.
- BERTRAND, J. W.; FRANSOO, J. C. Operations management research methodologies using quantitative modeling. *International Journal of Operations and Production Management*, v. 22, n. 2, p. 241-264, 2002. <http://dx.doi.org/10.1108/01443570210414338>
- BOWKER, J. *Jato comercial da Rússia faz primeiro voo público na Paris Air Show*. 2009. Disponível em: <<http://noticias.uol.com.br/ultnot/economia/2009/06/15/ult29u68101.jhtm>>. Acesso em: 13 jul. 2009.
- BRASIL. Ministério do Desenvolvimento. Secretaria de Comércio Exterior - SECEX. *Estatísticas de comércio exterior*. 2009. Disponível em: <<http://www.desenvolvimento.gov.br/sitio/interna/interna.php?area=5&menu=1161>>. Acesso em: 30 jun. 2009.
- COFFMAN JUNIOR, E. G. *Computer and job shop scheduling theory*. New York: John Wiley, 1976.
- CONWAY, R. W.; MAXWELL, W. L.; MILLER, L. W. *Theory of scheduling*. Addison-Wesley, 1967.
- DUIN, C. W.; VAN DER SLUIS, E. On the complexity of adjacent resource scheduling. *Journal of Scheduling*, v. 9, p. 49-62, 2006. <http://dx.doi.org/10.1007/s10951-006-5593-6>
- EMBRAER. *Embraer executive jets programs advance*. 2008. Disponível em: <http://www.embraer.com.br/institucional/download/2_068-Com-VAE-Program-Updates_EBACE-I-08.pdf>. Acesso em: 20 jul. 2009.
- EMBRAER. *Programas Legacy 450 e Legacy 500 da Embraer avançam*. 2009. Disponível em: <http://www.embraer.com.br/institucional/download/1_062-Prd-VAE-L450-500_Advance-P-09.pdf>. Acesso em: 20 jul. 2009.
- HAX, A.; CANDEA, D. *Production inventory management*. New Jersey: Prentice-Hall, 1984.
- HEIKE, G. et al. Mixed model assembly alternatives for low-volume manufacturing: the case of the aerospace industry. *International Journal of Production Economics*, v. 72, p.103-120, 2001. [http://dx.doi.org/10.1016/S0925-5273\(00\)00089-X](http://dx.doi.org/10.1016/S0925-5273(00)00089-X)
- HOWE, D. *Aircraft loading and structural layout*. Reston, Virginia: American Institute of Aeronautics Inc - AIAA, 2004. p. 391-422.
- LAWLER, E. L. et al. Sequencing and scheduling: algorithms and complexity. In: GRAVES, S. C.; RINNOOY KAN, A. H. G.; ZIPIKIN, P. H. (Ed.). *Logistics of production and inventory*. North-Holland: Elsevier, 1993. p. 455-522. *Handbooks in Operations Research and Management Science*, v. 4.
- LEUNG, J. Y-T. *Handbook of scheduling: algorithms, models and performance analysis*. Boca Raton: CRC Press, 2004.
- MEGSON, T. H. G. *Aircraft structures for engineering students*. 3. ed. Amsterdam: Elsevier Butterworth-Heinemann, Oxford, 1999. p. 211-232.
- MORABITO, R.; PUREZA, V. Modelagem e simulação. In: CAUCHICK, P. (Ed.). *Metodologia de pesquisa em engenharia de produção*. Rio de Janeiro: Campus/Elsevier, 2009.
- MORTON, T. E.; PENTICO, D. W. *Heuristic scheduling systems: with applications to production systems and project management*. New York: John Wiley & Sons, 1993.
- MURRAY, A. Spatial restrictions in harvest scheduling. *Forest Sciences*, v. 45, n. 1, p. 1-8, 1999.
- NAHMAS, S. *Production and operation analysis*. 4 ed. Boston: McGraw-Hill, 2001.
- NIU, M. C. Y. *Airframe structural design: practical design information and data on aircraft structures*. Hong Kong: Hong Kong Conmilit Press, 1988.
- OLMOS, M. *Presidente da Airbus planeja parceria com a Embraer*. 2008. Disponível em: <http://www.defesanet.com.br/eads/br_10.htm>. Acesso em: 13 jul. 2009.
- OTTOBONI, J. *China endurece para comprar da Embraer*. 2007. Disponível em: <<http://www.defesanet.com.br/emb1/cn.htm>>. Acesso em: 13 jul. 2009.
- PINEDO, M. *Planning and scheduling in manufacturing and services*. New York: Springer, 2005.
- PINEDO, M. *Scheduling: theory, algorithms, and systems*. 2 ed. New Jersey: Prentice Hall, 2008.
- POTTS, C. N.; STRUSEVICH, V. A. Fifty years of scheduling: a survey of milestones. *Journal of the Operational Research Society*, v. 60, p. 41-68, 2009. <http://dx.doi.org/10.1057/jors.2009.2>
- SANTOS, L. R. et al. Um Modelo para a programação de rotações de culturas. *Pesquisa Operacional*, v. 27, n. 3, 2007.
- SANTOS, L. R. et al. Crop rotation scheduling with adjacency constraints. *Annals of Operations Research*, p. 1-16, 2008.
- SCHMENNEN, R. W. *Production, operations management - Concepts and situation*. 4 ed. New York: MacMillan Publishing Company, 1990. p. 674-705.
- SUKHOI. *Komsomolsk-on-Amur hosted the rollout of the first Sukhoi Superjet 100 ready to fly*. 2007. Disponível em : <<http://www.sukhoi.org/eng/news/company/?id=1306>>. Acesso em: 13 jul. 2009.
- WEINTRAUB, A.; MURRAY, A. T. Review of combinatorial problem induced by spatial forest harvesting planning. *Discrete Applied Mathematics - IV ALIO/EURO Workshop on Applied Combinatorial Optimization*, v. 154, n. 5, p. 867-879, 2006.

Production scheduling of assembly fixtures with adjacent constraints in the aeronautics industry

Abstract

This paper deals with a production scheduling problem that typically occurs in the aeronautics industry involving special structures called assembly fixtures, composed of several workstations in parallel, which are used to assemble the parts of the aircrafts. Tasks should be scheduled to be performed in these workstations in order to minimize the make-span; however, in addition to the usual constraints, such as due dates and tasks priority, there are also limitations that prevent two tasks from being performed at the same time, in two adjacent workstations on the assembly fixture. A mixed integer model to represent the problem was proposed based on practical case studies of assembly fixtures schedules of an aeronautic company. The solutions generated by the model were implemented, in practice, with gains both in the use of the assembly fixtures studied, as well as in the use of the required labor.

Keywords

Production scheduling. Assembly fixtures. Aeronautics industry. Mixed integer programming.

Anexo 1. Detalhamento dos conjuntos W, A e B.

Exemplo 2

W: Conjunto das tarefas que utilizam o mesmo posto de trabalho:

$W = \{(1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (5, 6), (5, 7), (5, 8), (6, 7), (6, 8), (7, 8), (13, 14), (13, 15), (13, 16), (13, 17), (13, 18), (14, 15), (14, 16), (14, 17), (14, 18), (15, 16), (15, 17), (15, 18), (16, 17), (16, 18), (17, 18), (19, 20), (19, 21), (19, 22), (19, 23), (19, 24), (20, 21), (20, 22), (20, 23), (20, 24), (21, 22), (21, 23), (21, 24), (22, 23), (22, 24), (23, 24), (25, 26), (25, 27), (25, 28), (26, 27), (26, 28), (27, 28), (29, 30), (29, 31), (29, 32), (30, 31), (30, 32), (31, 32), (33, 34), (33, 35), (33, 36), (34, 35), (34, 36), (35, 36)\}.$

A: conjunto de tarefas que possuem precedência entre si.

$A = \{(1, 2), (3, 4), (5, 6), (7, 8), (9, 10), (11, 12), (13, 14), (14, 15), (16, 17), (17, 18), (19, 20), (20, 21), (22, 23), (23, 24), (25, 26), (27, 28), (29, 30), (31, 32), (33, 34), (35, 36)\}.$

B: conjunto de tarefas que utilizam postos de trabalho adjacentes.

$B = \{(1, 5), (1, 6), (1, 7), (1, 8), (2, 5), (2, 6), (2, 7), (2, 8), (3, 5), (3, 6), (3, 7), (3, 8), (4, 5), (4, 6), (4, 7), (4, 8), (5, 9), (5, 10), (5, 11), (5, 12), (6, 9), (6, 10), (6, 11), (6, 12), (7, 9), (7, 10), (7, 11), (7, 12), (8, 9), (8, 10), (8, 11), (8, 12), (9, 13), (9, 14), (9, 15), (9, 16), (9, 17), (9, 18), (10, 13), (10, 14), (10, 15), (10, 16), (10, 17), (10, 18), (11, 13), (11, 14), (11, 15), (11, 16), (11, 17), (11, 18), (12, 13), (12, 14), (12, 15), (12, 16), (12, 17), (12, 18), (13, 19), (13, 20), (13, 21), (13, 22), (13, 23), (13, 24), (14, 19), (14, 20), (14, 21), (14, 22), (14, 23), (14, 24), (15, 19), (15, 20), (15, 21), (15, 22), (15, 23), (15, 24), (16, 19), (16, 20), (16, 21), (16, 22), (16, 23), (16, 24), (17, 19), (17, 20), (17, 21), (17, 22), (17, 23), (17, 24), (18, 19), (18, 20), (18, 21), (18, 22), (18, 23), (18, 24), (19, 25), (19, 26), (19, 27), (19, 28), (20, 25), (20, 26), (20, 27), (20, 28), (21, 25), (21, 26), (21, 27), (21, 28), (22, 25), (22, 26), (22, 27), (22, 28), (23, 25), (23, 26), (23, 27), (23, 28), (24, 25), (24, 26), (24, 27), (24, 28), (25, 29), (25, 30), (25, 31), (25, 32), (26, 29), (26, 30), (26, 31), (26, 32), (27, 29), (27, 30), (27, 31), (27, 32), (28, 29), (28, 30), (28, 31), (28, 32), (29, 33), (29, 34), (29, 35), (29, 36), (30, 33), (30, 34), (30, 35), (30, 36), (31, 33), (31, 34), (31, 35), (31, 36), (32, 33), (32, 34), (32, 35), (32, 36)\}.$

Exemplo 3

W: Conjunto das tarefas que utilizam o mesmo posto de trabalho:

$W = \{(1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4), (5, 6), (5, 7), (5, 8), (6, 7), (6, 8), (7, 8), (9, 10), (11, 12), (11, 13), (11, 14), (12, 13), (12, 14), (13, 14), (15, 16), (15, 17), (15, 18), (16, 17), (16, 18), (17, 18), (19, 20), (19, 21), (19, 22), (20, 21), (20, 22), (21, 22), (23, 24), (23, 25), (23, 26), (24, 25), (24, 26), (25, 26), (27, 28), (27, 29), (27, 30), (28, 29), (28, 30), (29, 30)\}.$

A: conjunto de tarefas que possuem precedência entre si.

$A = \{(1, 2), (3, 4), (5, 6), (7, 8), (9, 10), (11, 12), (13, 14), (15, 16), (17, 18), (19, 20), (21, 22), (23, 24), (25, 26), (27, 28), (29, 30)\}.$

B: conjunto de tarefas que utilizam postos de trabalho adjacentes.

$B = \{(1, 5), (1, 6), (1, 7), (1, 8), (2, 5), (2, 6), (2, 7), (2, 8), (3, 5), (3, 6), (3, 7), (3, 8), (4, 5), (4, 6), (4, 7), (4, 8), (5, 9), (5, 10), (6, 9), (6, 10), (7, 9), (7, 10), (8, 9), (8, 10), (9, 11), (9, 12), (9, 13), (9, 14), (10, 11), (10, 12), (10, 13), (10, 14), (11, 15), (11, 16), (11, 17), (11, 18), (12, 15), (12, 16), (12, 17), (12, 18), (13, 15), (13, 16), (13, 17), (13, 18), (14, 15), (14, 16), (14, 17), (14, 18), (15, 19), (15, 20), (15, 21), (15, 22), (16, 19), (16, 20), (16, 21), (16, 22), (17, 19), (17, 20), (17, 21), (17, 22), (18, 19), (18, 20), (18, 21), (18, 22), (19, 23), (19, 24), (19, 25), (19, 26), (20, 23), (20, 24), (20, 25), (20, 26), (21, 23), (21, 24), (21, 25), (21, 26), (22, 23), (22, 24), (22, 25), (22, 26), (23, 27), (23, 28), (23, 29), (23, 30), (24, 27), (24, 28), (24, 29), (24, 30), (25, 27), (25, 28), (25, 29), (25, 30), (26, 27), (26, 28), (26, 29), (26, 30)\}.$