

Programação da Produção de Gabaritos de Montagem com Restrições de Adjacência na Indústria Aeronáutica

Bruno Jensen Virginio da Silva
Reinaldo Morabito
Departamento de Engenharia de Produção
Universidade Federal de São Carlos

Horacio Hideki Yanasse
Laboratório Associado de Computação e Matemática Aplicada
Instituto Nacional de Pesquisas Espaciais

Resumo: Esse artigo trata de um problema de programação da produção que aparece tipicamente na indústria aeronáutica, envolvendo estruturas especiais de montagem chamadas gabaritos, compostas de diversos postos de trabalho em paralelo para montar 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. Propõe-se um modelo de programação linear inteira mista para representar o problema, com base no estudo de casos práticos de programação de gabaritos de montagem presentes em uma indústria aeronáutica. As soluções geradas pelo modelo foram implementadas na prática com ganhos significativos, tanto na utilização dos gabaritos de montagem estudados, quanto na utilização de mão-de-obra envolvida.

Palavras-chave: programação da produção, gabaritos de montagem, indústria aeronáutica, programação linear inteira mista.

Production Scheduling of Assembly Fixtures with Adjacent Constraints in the Aeronautics Industry.

Abstract: This paper deals with a production scheduling problem that typically appears in the aeronautics industry, involving special structures called assembly fixtures, composed of several workstations in parallel to assemble parts of the aircrafts. Tasks should be scheduled to run in these workstations in order to minimize the makespan; however, beyond the usual

constraints, such as due dates and precedence restraints among tasks, there are also constraints that prevent two tasks can be performed at same time in two adjacent workstations on the assembly fixture. It is proposed a mixed integer model to represent the problem, based on case studies of schedules of assembly fixtures present in an aeronautics industry. The solutions generated by the model were implemented in practice with significant gains in both the use of assembly fixtures studied, as in the use of labor required.

Key-words: production scheduling, assembly fixtures, aeronautics industry, mixed integer programming.

1. Introdução

No Brasil, uma indústria que vem crescendo de maneira significativa nas últimas décadas é a indústria aeronáutica. Nos últimos oito anos, a principal empresa deste segmento tem ocupado os primeiros lugares na lista de empresas exportadoras brasileiras. Em 2008, por exemplo, ela 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 crescente participação com a entrada em novos mercados de aviação (EMBRAER, 2008, 2009) e o lançamento de novos produtos, aumentando ainda mais a sua importância para a economia brasileira. Neste contexto, estudos de novas formas de melhorar a eficiência são particularmente relevantes, principalmente com o ingresso de novos concorrentes diretos no mercado de aviação comercial (OTTOBONI, 2007; SUKHOI, 2007; OLMOS, 2008; BOWKER, 2009).

Grosso modo, a produção de aviões segue basicamente duas etapas: fabricação e montagem. A montagem pode ser vista sob vários aspectos, desde as montagens de pequenos subconjuntos até as montagens estruturais dos aviões propriamente. Nas operações de montagem, uma tecnologia comumente utilizada é o uso de gabaritos de montagem. Os gabaritos são a forma que a indústria aeronáutica utiliza para garantir as condições de projeto dos produtos montados e a sua intercambiabilidade com outras partes que se juntam em montagens posteriores. Eles variam em tamanho e complexidade, dependendo das peças e operações que nele são realizadas. Geralmente, as operações de montagem realizadas nos gabaritos são manuais, sendo os operadores os responsáveis pelo posicionamento e montagens 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 aos produtos neles montados; logo, a quantidade de aviões a ser produzida depende da utilização dos gabaritos de montagem, podendo restringir a capacidade de produção. 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 segmentos parecidos de 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, esse último tipo de gabarito é considerado, com a particularidade de que quando alguns produtos são montados usando alguns postos de trabalho do gabarito, outros produtos montados em postos de trabalho adjacentes não podem ser montados ao mesmo tempo. Essa limitação é aqui chamada de restrição de adjacência. Convém destacar que não foram encontrados estudos que tratam especificamente desta programação da produção de máquinas com restrições de adjacência na literatura clássica de teoria de *scheduling* (e.g., CONWAY et al., 1967; BAKER, 1974; COFFMAN, 1976; HAX; CANDEA, 1984; MORTON; PENTICO, 1993; NAHMIAS, 2001, PINEDO, 2005, 2008) nem nas bases de dados da literatura especializada de gestão da produção e pesquisa operacional, o que sugere que esse problema ainda não foi abordado na literatura e nem pode ser classificado diretamente conforme as classes de problemas de programação de Rodamer e White (1988). 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, mas cujas abordagens não podem ser diretamente aplicadas neste estudo.

O objetivo do presente trabalho é estudar o problema de programação da produção de gabaritos de montagem com restrições de adjacência, focando o estudo em um tipo específico encontrado no processo de montagem de jatos comerciais de uma indústria aeronáutica. Apresenta-se um modelo de programação linear inteira mista para representar o problema, que foi implementado computacionalmente em linguagem de modelagem GAMS e resolvido utilizando o *software* de otimização CPLEX. Acredita-se que, com pequenas modificações, o modelo também possa ser adaptado para representar a programação da produção de outros gabaritos de montagem desta e de outras empresas aeronáuticas. Trata-se, portanto, de uma pesquisa baseada em modelagem quantitativa com abordagem de método 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 mais 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 descrito 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, grosso modo, em fabricação e montagem, e o problema estudado é um problema de montagem. Problemas de montagem podem ser separados em três tipos: montagem de subconjuntos, montagem estrutural e montagem final. A montagem de *subconjuntos* nada mais é do que qualquer montagem que envolve 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 são compostos de várias peças, utilizam gabaritos de montagem, possuem tamanhos maiores, variando entre cerca de 1 e 3 metros, e são fornecidos diretamente para uma etapa de montagem estrutural ou montagem final. Esse é o caso do problema estudado.

A montagem estrutural é aquela que monta as partes estruturais do avião, como a fuselagem, asas e empenagem. A montagem final é a etapa na qual 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 até o vôo de produção e entrega ao cliente. Os detalhes sobre as estruturas dos aviões, sua complexidade e um pouco das 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 ser executada. No caso estudado, 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 partes, não havendo nenhum detalhamento maior. O detalhamento da programação é responsabilidade da 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 produzidas são conhecidas porque são baseadas nas cadências de produção dos aviões. Estas cadências nada mais são do que a quantidade mensal de aviões que será produzida de certo modelo. 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 em alguns momentos da sua história recente foram da ordem de 20 aviões por mês. Alguns modelos de aeronaves produzidas, que são os jatos comerciais, utilizam dois subconjuntos 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. A figura 1 ilustra um subconjunto composto de oito *partes* (denominadas partes 1, 2, ..., 8). Cada parte possui apenas um *posto de trabalho* específico no gabarito para ser montada, mostrada na figura 2. A parte 1 do subconjunto deve ser montada no posto de trabalho 1 do gabarito, a parte 2 deve ser montada no posto de trabalho 2, e assim sucessivamente, para todas as 8 partes do subconjunto.

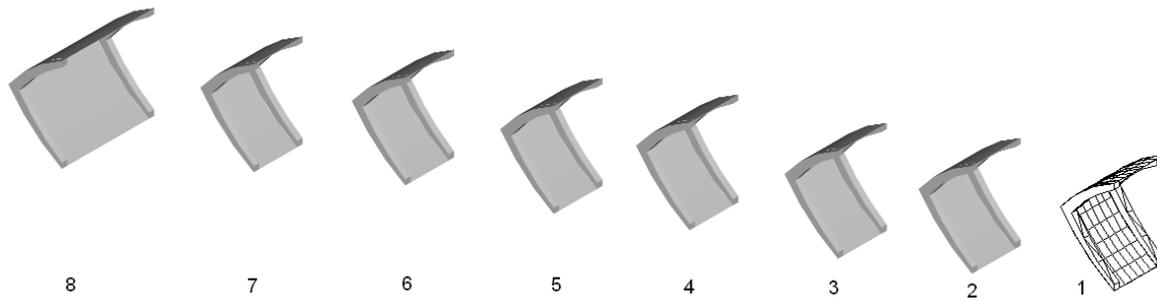


Figura 1: Subconjunto e as suas oito partes

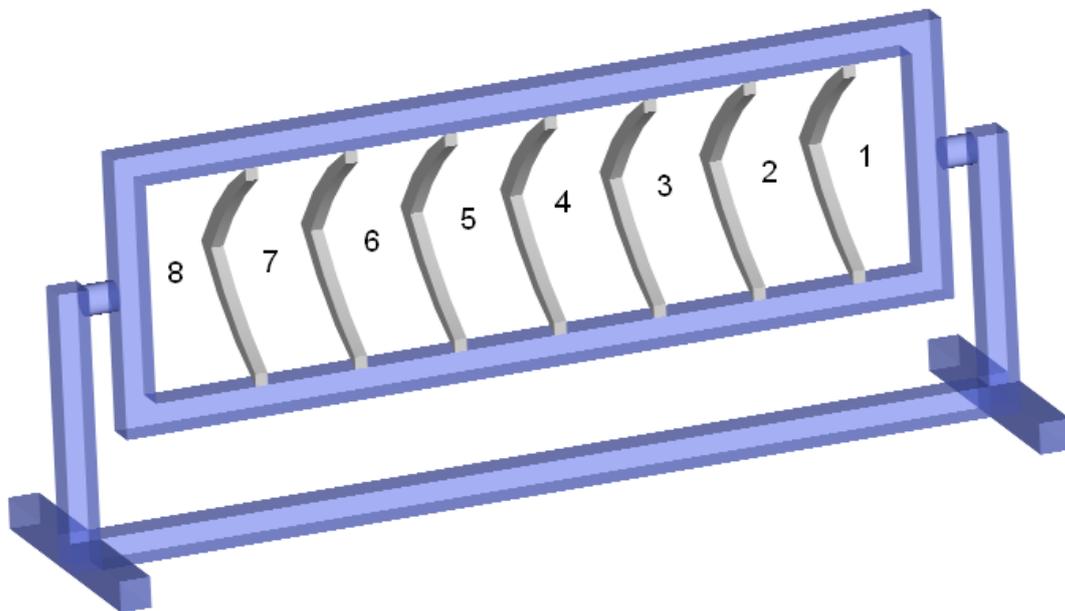


Figura 2: Gabarito de montagem do subconjunto com oito postos de trabalho

As montagens das oito partes do subconjunto são realizadas individualmente. Cada parte 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 parte do subconjunto. As oito partes do subconjunto são montadas individualmente no gabarito e são, posteriormente, montadas na forma de subconjunto apenas em outra etapa de montagem que não é considerada nesse trabalho. Isso significa que cada parte do subconjunto pode ser montada e entregue individualmente, mesmo fazendo parte do mesmo avião. A administração das partes consiste basicamente na administração dos prazos de entrega de cada parte, 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 partes do subconjunto que utilizam postos de trabalho (estações) adjacentes. Por exemplo, se um operador está montando a parte 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 partes do subconjunto que neles devem ser montadas.

As montagens de cada parte acontecem 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 partes vão para as bancadas, para os processos de furação e rebitagem, que não precisam do gabarito para ser executados, já que o posicionamento dos componentes já foi feito. A segunda etapa se inicia novamente no gabarito, terminando com uma complementação novamente realizada nas bancadas. Isso ocorre por restrições técnicas, pois nem todos os componentes podem ser totalmente referenciados na primeira etapa do gabarito; então é necessária uma segunda etapa para a maioria das partes. Em alguns casos pode-se ter uma terceira etapa de montagem da parte, também composta de uma operação no gabarito, seguida de uma operação nas bancadas.

Cada par de operações no gabarito e na bancada forma então uma *tarefa*. A figura 3 representa uma rede de precedência das montagens de cada parte - cada uma composta de apenas duas tarefas. Em branco são as operações executadas no gabarito e em cinza, as operações executadas na bancada. Os números dentro da atividade (1 a 16) identificam as tarefas associadas às montagens das partes do subconjunto e às operações de montagem no gabarito (1) ou nas bancadas (2).

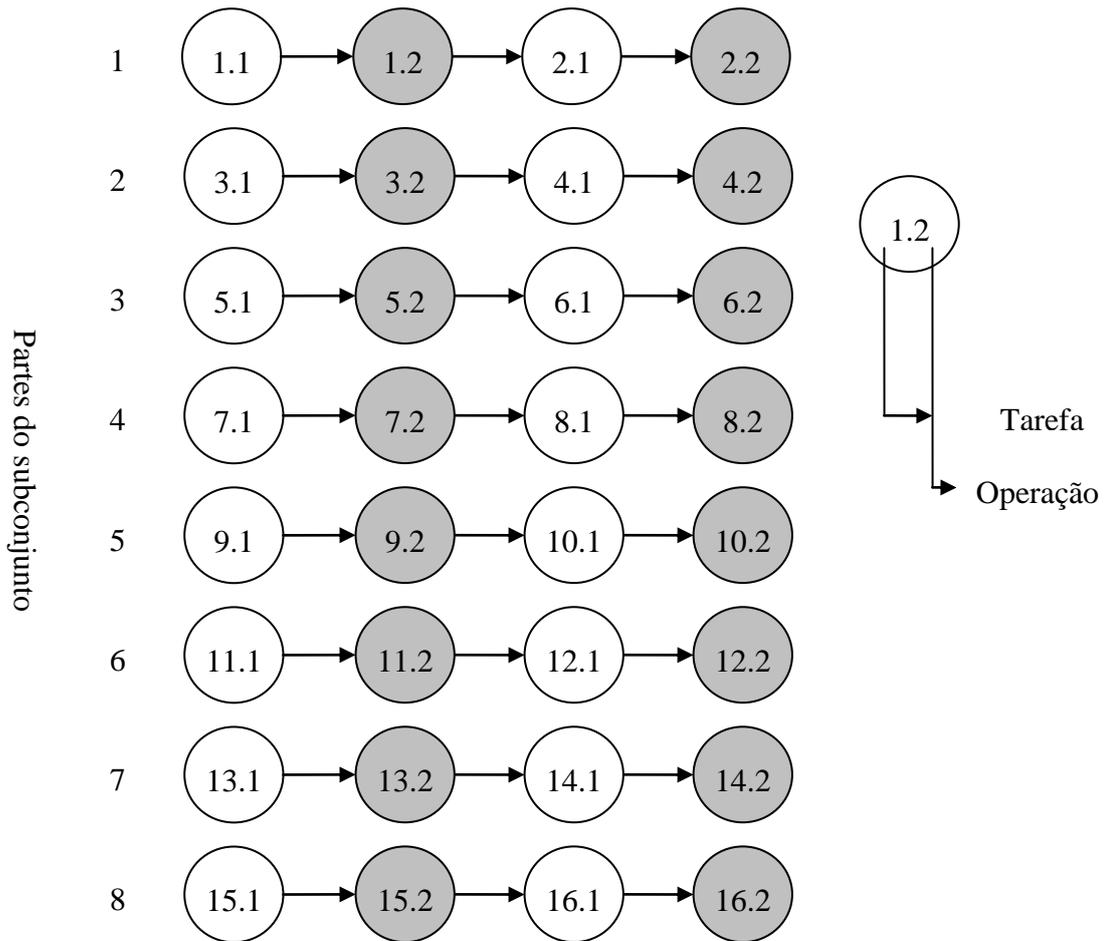


Figura 3: Rede de precedência da montagem do subconjunto

Perceba que, nesta rede da figura 3, cada parte do subconjunto é formada por duas tarefas e cada tarefa possui duas operações. Por exemplo, a parte 1 do subconjunto é formada pelas tarefas 1 e 2. A tarefa 1 é composta das operações 1 e 2 e, também, a tarefa 2 é composta das operações 1 e 2. Assim, a figura 3 mostra as outras relações entre parte do subconjunto, tarefa e operação para todas as partes do subconjunto. Em branco estão as operações executadas no gabarito e em cinza estão as operações de bancada.

O gráfico de Gantt do programa de produção de um subconjunto tem um padrão bastante singular. Na figura 4 pode ser visto um exemplo de programação da produção das primeiras três partes e seis tarefas da figura 3. A aparência do gráfico é justamente de um padrão com diversos espaços em branco, que significam postos de trabalho sem montagens ou com montagens em postos de trabalho adjacentes, o que acaba bloqueando as tarefas a serem executadas e criando os espaços sem tarefas.

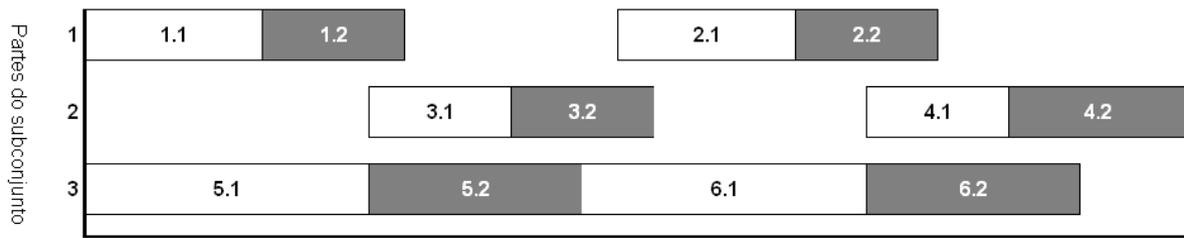


Figura 4: Gráfico de Gantt de uma programação para o gabarito

Note na figura 4 que as operações em branco são as montagens no gabarito e as operações em cinza são as montagens nas bancadas. Repare 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 que a tarefa 1 tiver as duas operações terminadas. O mesmo acontece com as tarefas 3 e 4, e as tarefas 5 e 6. Perceba que a programação da figura 4 respeita essas restrições.

3. Modelagem Matemática

Este problema pode ser visto como um problema de programação de máquinas paralelas distintas, porém, dependentes. O problema é basicamente programar a montagem das partes do subconjunto no gabarito, com o objetivo de minimizar o tempo total para completar as tarefas (*makespan*). Deve considerar as restrições de adjacência, além das restrições de prazos de disponibilidade e de entrega das tarefas, as restrições de precedência entre as tarefas de montagem e as restrições de tecnologia (i.e., o posto de trabalho do gabarito em que cada tarefa deve ser executada). As variáveis de decisão do modelo consideram cada tarefa e, indiretamente, a operação de cada tarefa.

Muitos problemas de programação da produção são NP-difícil (LAWLER et al., 1993; LEUNG, 2004). Nesta seção é apresentado um modelo matemático linear inteiro misto para representar o problema de programação da produção de um gabarito com restrições de adjacência, cuja solução não nos parece ser imediata ou trivial. Mostrar a complexidade deste problema está além do escopo deste trabalho e é um tópico para pesquisa futura. Sejam os seguintes parâmetros do modelo:

- p_j duração da operação no gabarito da tarefa j ($j = 1, 2, \dots, J$)
- q_j duração da operação na bancada 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 disso, 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

O problema é minimizar o *makespan* t_F , sujeito a restrições de disponibilidade e prazos de entrega das tarefas, restrições de seqüenciamento das tarefas em cada posto de trabalho do gabarito, restrições de precedência entre as tarefas e restrições de adjacência, ou seja:

$$\text{Min } t_F \tag{1}$$

$$t_j + p_j + q_j \leq t_F \quad j = 1, \dots, J \tag{2}$$

$$t_j \geq r_j \quad j = 1, \dots, J \tag{3}$$

$$t_j + p_j + q_j \leq d_j \quad j = 1, \dots, J \tag{4}$$

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

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

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

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

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

$$t_j \geq 0, \quad j = 1, \dots, J, \quad y_{jk} \in \{0,1\}, \quad j = 1, \dots, J, \quad k = 1, \dots, J, \quad j \neq k \tag{10}$$

A função objetivo junto com a restrição (2) garante que o *makespan* t_F 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 p_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 - 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: $\sum_{j=1}^J (p_j + q_j)$.

As restrições (7) garantem a relação de precedência 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 operações 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.

Observe que, por meio do modelo (1)-(10), é possível realizar experimentos computacionais, por exemplo, para estimar a cadência máxima desta unidade de produção e suas necessidades de mão-de-obra. Estes experimentos podem fornecer subsídios importantes ao decisor no planejamento de expansão de capacidade e recursos humanos nos gabaritos em casos como, por exemplo, aumentos das vendas.

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 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 um problema exemplo 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,...,6) precisa ser programada neste gabarito. As durações (em alguma unidade de tempo) das operações e os respectivos postos de trabalho das tarefas são mostrados na tabela 1.

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

Parte do subconjunto	Tarefa (j)	Posto de trabalho no gabarito	Tempo no gabarito (p_j)	Tempo na bancada (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

Nesse caso, o subconjunto ilustrativo é formado por três partes, onde cada parte possui apenas duas tarefas, cada uma com duas operações, ou seja, um total de quatro operações. Note na tabela 1 que existem 6 tarefas e 12 operações. As tarefas 1 e 2 são executadas no mesmo posto de trabalho. O mesmo vale para as tarefas 3 e 4, e 5 e 6. Considera-se que todas as tarefas estão disponíveis no instante zero (i.e., $r_j = 0$), e que o instante 50 é o prazo de entrega de todas as tarefas (i.e., $d_j = 50$).

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: $B = \{(1, 3), (1, 4), (2, 3), (2, 4), (3, 5), (3, 6), (4, 5), (4, 6)\}$.

A figura 5 apresenta o gráfico de Gantt obtido 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 indicam as tarefas e, os gráficos das operações de gabarito (p_j) e de operações da bancada (q_j) estão mostrados 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.

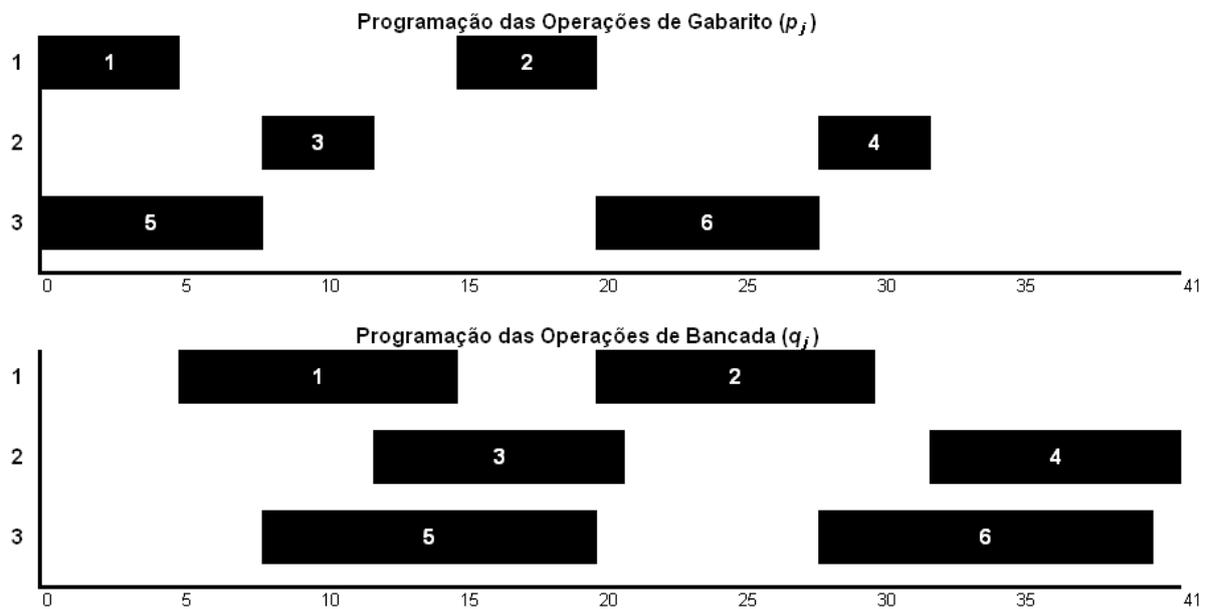


Figura 5: Gráfico de Gantt com a solução para o exemplo ilustrativo (os números indicam as tarefas)

Também existem intervalos de tempo em que o gabarito não possui tarefas em processamento porque está esperando o encerramento de operações em processamento nas bancadas, como é o caso logo após o término da operação da tarefa 3 no gabarito. A operação da tarefa 3 executada na bancada possui uma folga, podendo ser iniciada dentro de um intervalo de tempo que não compromete a solução ótima apresentada. Note que, apesar de todas as tarefas estarem disponíveis para serem iniciadas no instante zero, a tarefa 3 precisa esperar para ser iniciada. Além disso, a tarefa 4 também fica aguardando processamento por certo período. O tempo total para completar as tarefas, *makespan*, é igual a 41. Como o prazo de entrega de todas as tarefas é 50, todas elas terminam dentro do prazo.

Embora este exemplo ilustrativo seja relativamente simples, perceba que sua solução não é trivial. A dificuldade de determinar um programa viável cresce com o aumento dos números de tarefas e postos de trabalho do gabarito.

4.2 Exemplos Reais

Nessa subseção são analisados dois exemplos reais (exemplos 2 e 3) da empresa aeronáutica estudada. No exemplo 2, das oito partes que formam um subconjunto, duas partes

são montadas em três etapas envolvendo tarefas com 6 operações intercaladas de montagem, três de gabarito e três de bancadas, conforme apresentado na tabela 5 (veja as 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. Também a tarefa 16 deve preceder a tarefa 17, que por sua vez deve preceder a tarefa 18.

Na época em que os dados deste exemplo foram coletados, a produção de aviões que utilizavam este subconjunto estava na sua maior cadência de produção até então, e o total de aeronaves produzidas era menor do que 100 aviões. Portanto, pode-se dizer para o contexto da indústria aeronáutica e para um avião de médio porte que se tratava de um modelo de avião ainda em fase inicial de produção. A implicação disso é que os dados estão sujeitos a maiores mudanças devido a evolução da curva de aprendizagem (SCHMENNER, 1990).

Um subconjunto completo, no caso do exemplo 2, é composto por 18 tarefas e suas operações podem ter durações diferentes. 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. Conforme discussão feita em mais detalhes na seção 5, a programação de dois subconjuntos é a quantidade conveniente para a programação das montagens na prática. Portanto, o exemplo 2 trata de um total de 36 tarefas, número equivalente a dois subconjuntos e equivalente à quantidade de subconjuntos que cada produto final precisa.

Na tabela 5 são apresentados os dados das tarefas. Perceba que existem tarefas idênticas, decorrentes de partes idênticas, mas referentes a cada um dos dois subconjuntos considerados no exemplo. Os conjuntos *W*, *A* e *B* estão detalhados no anexo.

Tabela 5: Duração (em u.t.) das operações das partes do subconjunto para o exemplo 2

Subconjunto	Parte do subconjunto	Tarefa (j)	Posto de trabalho no gabarito	Tempo no gabarito (p_j)	Tempo na bancada (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
		15	4	45	60	14
2	4	16	4	90	150	-
		17	4	90	150	16
		18	4	45	60	17
1	5	19	5	90	150	-
		20	5	90	150	19
		21	5	45	60	20
2	5	22	5	90	150	-
		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

O modelo do 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. A figura 6 mostra os gráficos de Gantt obtidos a partir da solução deste problema. O primeiro gráfico é o programa das operações do gabarito e o segundo gráfico é 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.

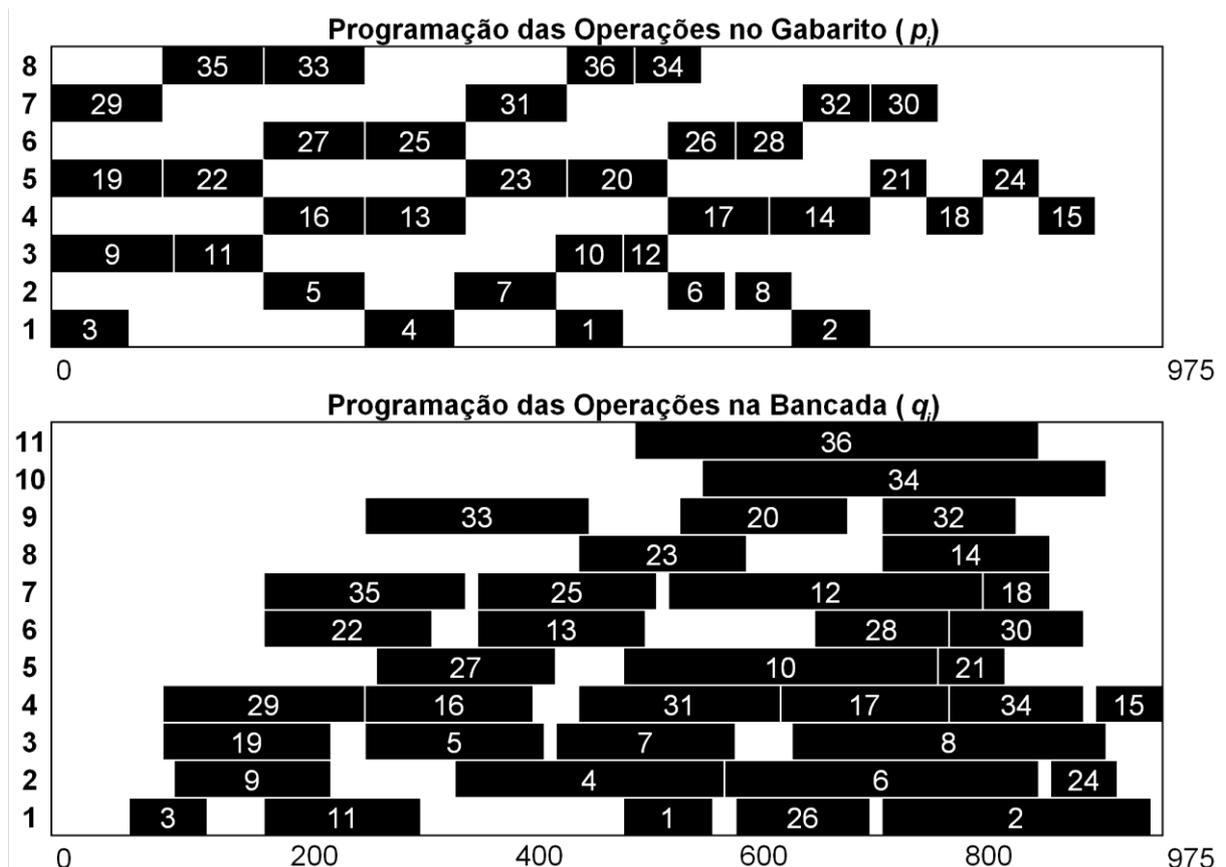


Figura 6: Gráfico de Gantt da programação das operações de gabarito e de bancada do exemplo 2

Todas as operações nos postos de trabalho do gabarito não possuem operações adjacentes executadas ao mesmo tempo, conforme pode ser verificado no primeiro gráfico da figura 6. No segundo gráfico de Gantt da figura 6 é apresentado o programa de produção das operações executadas nas bancadas. Embora nas bancadas existam apenas as restrições de precedência entre as operações (não ocorrendo as restrições de adjacência e de tecnologia), o programa gera também um padrão com espaços em branco, ou seja, sem operações para serem executadas.

Nas bancadas 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. No início da programação, a quantidade de tarefas em

processamento é menor por conta das tarefas que eventualmente as precedem estarem com as operações sendo executadas no gabarito. O inverso acontece no final da programação, em que há uma quantidade maior de tarefas em processamento e um número menor de operações no gabarito.

A seguir analisam-se os resultados da aplicação do modelo para resolver o exemplo 3. As diferenças entre os exemplos 2 e 3 são, basicamente, que no exemplo 3 todas as tarefas possuem no máximo 4 operações de montagem (no gabarito e nas bancadas), enquanto no exemplo 2 as partes 4 e 5 possuem 6 operações de montagem. Além disso, algumas durações de operações são diferentes, diminuindo a duração ou transferindo partes do processo entre as operações. Essas diferenças são decorrentes de evoluções técnicas do processo de montagem efetuadas pela engenharia de processos, mas que mudam os dados e a solução do problema de modo considerável.

Tabela 6: Duração em unidades de tempo das partes do subconjunto para o modelo 3

Subconjunto	Parte do subconjunto	Tarefa (j)	Posto de trabalho no gabarito	Tempo no gabarito (p_j)	Tempo na bancada (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	-
2	3	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

A tabela 6 mostra as durações das operações e os respectivos nomes (j) das tarefas do exemplo 3. Nesse caso, são necessárias 30 tarefas para montar as partes de dois subconjuntos inteiros, seis tarefas a menos que no modelo 2. A parte 3 do subconjunto possui apenas uma tarefa nesse caso (tarefas 9 e 10, uma para cada um dos dois subconjuntos). Os respectivos conjuntos W , A e B estão detalhados no anexo.

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, *makespan*, foi de 815 u.t. As operações do gabarito terminam no tempo 615 u.t. Comparando-se os programas mostrados nas figuras 6 e 7, percebe-se que embora os problemas sejam similares, a mudança no número de tarefas e na duração delas gera programas totalmente diferentes.

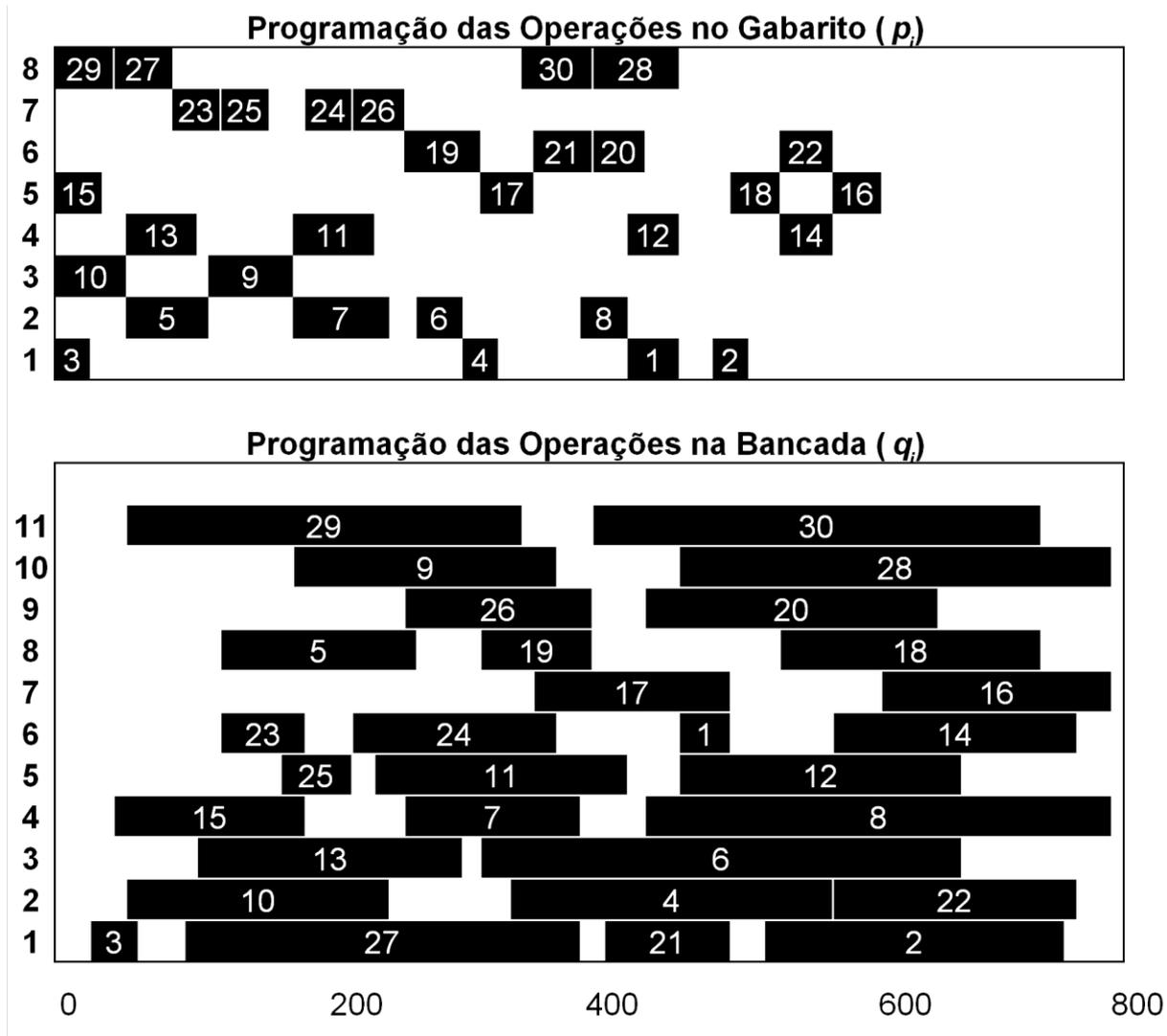


Figura 7: Gráfico de Gantt da programação do gabarito do modelo 3

O *makespan* para o exemplo 2 foi de 915 u.t. e do exemplo 3 foi 810 u.t. A diminuição do *makespan* no exemplo 3 se deve à redução do número de tarefas e na duração total da montagem de cada parte do subconjunto verificada em todas as partes, com exceção da parte 2 e da parte 8 do subconjunto. Cada uma das montagens realizada nas bancadas, mostrada nas figuras 6 e 7, precisa de uma pessoa. Assim, o número de operações de montagens ocorrendo ao mesmo tempo (i.e., em paralelo no diagrama de Gantt) mostra o número de pessoas necessárias nas bancadas para efetuar a montagem. Note que o número máximo de pessoas utilizado no exemplo 2 foi 10 e no exemplo 3 foi 11, para as operações de bancada.

5. Implementação das Soluções 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 intuitivo. A política aplicada era

estabelecer prazos de entrega para os subconjuntos (sugeridos pelo sistema de *Materials Requirement Planning* - MRP - da empresa) e montar primeiro as partes dos subconjuntos que possuíssem prazo de entrega mais curto, ou que estivessem mais atrasadas (EDD, *earliest due date*). A escolha da abordagem não foi feita considerando as suas vantagens e desvantagens, e a consequência é que havia um seqüenciamento das partes dos subconjuntos a serem montadas, com horizonte de no máximo um dia, e não uma programação de tarefas. Esse modo de organizar a montagem é chamado neste trabalho de seqüenciamento por parte.

Outro modo de organizar a montagem seria a programação por avião. Conhecendo-se a cadência de produção dos aviões, pode-se calcular o tempo de ciclo meta necessário para atender à demanda. Por exemplo, para o caso de uma cadência de 10 aviões/mês e um mês com 20 dias úteis, então o tempo de ciclo é:

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

Assim, cada par de subconjuntos dos 10 aviões seria montado a cada dois dias do mês. A figura 8 ilustra os programas de produção de cada uma das duas formas de programar a produção. O número dentro da tarefa identifica o número de série do avião a ser montado. Em cinza estão os finais de semana. No primeiro programa, como a montagem é feita parte por parte, considera-se que a tarefa de montagem do avião inicia quando começa a montagem da primeira de qualquer uma das partes do par de subconjuntos, e termina quando a última parte está pronta.

Repare no gráfico de montagem de subconjuntos por parte na figura 8 que o avião 1 começa a produção no início do período e o avião 2 começa a montagem no período 2 e ambos duram 5 unidades de tempo. O avião 5, que é formado exatamente pelas mesmas operações, demora 7 unidades de tempo para ser montado, ou seja, a primeira operação de montagem ocorreu no momento 10 e a última no momento 18.

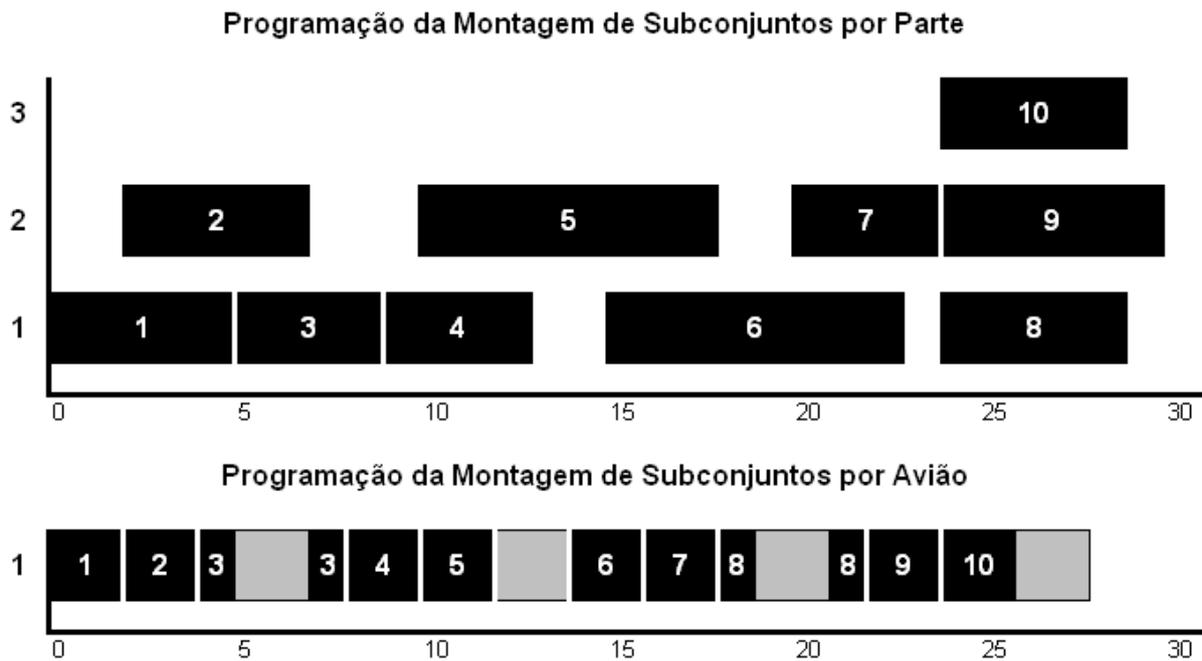


Figura 8: Programação por parte e programação por subconjunto

No segundo programa, é montado apenas um avião por vez, ou seja, um par de subconjuntos referentes a um avião em específico. Observando a figura 8, pode-se perceber que os aviões 1 e 2 são montados ao mesmo tempo no seqüenciamento por partes, mas não são montados ao mesmo tempo na programação por aviões. O mesmo acontece com os aviões 6 e 7 ou 8, 9 e 10. Comparando-se as duas formas, no primeiro caso, as montagens têm durações que variam para cada avião, mesmo sendo as mesmas montagens que do segundo caso. Portanto, existe uma incerteza maior na duração da montagem do primeiro programa do que do segundo. Isso é consequência da existência de tempos de espera, estoque em processo e ociosidade, que aumentam o *lead time* de produção. Para utilizar o segundo programa, seria necessário encontrar um programa de produção viável para montar um avião dentro do tempo de ciclo para uma dada cadência. Para isso, foi utilizado o modelo matemático apresentado nesse trabalho, e as soluções como as dos exemplos 2 e 3 anteriormente apresentados, podem então ser implantadas na produção.

As vantagens do segundo programa em relação ao primeiro são: o gerenciamento é mais fácil, a variabilidade na solicitação de fornecimento de peças componentes e na entrega de subconjuntos prontos é menor, o *lead time* de produção e o estoque em processo são menores, a quantidade de horas extras é menor, a utilização da mão-de-obra é melhor e proporciona um aumento gradativo da produtividade como consequência da repetição do mesmo programa.

6. Conclusões

O problema de programação da produção de gabaritos abordado nesse artigo aparentemente ainda não foi estudado e reportado na literatura, embora possa ter implicações práticas importantes na indústria aeronáutica, conforme apresentado. As soluções obtidas pelo modelo proposto mostraram vantagens em relação às soluções empregadas pela empresa, tanto na utilização dos gabaritos de montagem estudados, quanto na utilização de mão-de-obra envolvida. 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 pode fornecer subsídios importantes ao decisor no planejamento de capacidade e recursos humanos nos gabaritos e nas bancadas.

Em situações em que as soluções do problema de programação envolvam necessidades de mão-de-obra excessivas nas bancadas, uma perspectiva interessante para 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 violadas de mão-de-obra 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 (ou minimizar a mão-de-obra requerida), ao mesmo tempo restritos a um dado *makespan* escolhido em função do tempo de ciclo da cadência de produção.

Outra perspectiva interessante de pesquisa futura seria a adaptação e aplicação das abordagens na programação da produção de outros gabaritos, que apresentem restrições de adjacência similares às presentes na indústria aeronáutica, mas que possuam restrições ou funções objetivos diferentes do problema apresentado nesse trabalho, devido a, por exemplo, diferentes tecnologias e modos de execução empregados nestes gabaritos.

Agradecimentos: Os autores agradecem a Silvia Regina Morales por colaborar à pesquisa. Esta pesquisa contou com apoio financeiro da Fapesp e CNPq.

Anexo

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.

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

Referências Bibliográficas

ARENALES, M, ARMENTANO, V. A., MORABITO NETO, R., YANASSE, H. H. **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**. vol. 22, n.2, 2002, pág. 241-264, 2002.

BOWKER, J. **Jato Comercial da Rússia faz primeiro vôo público na Paris Air Show**. 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**. Disponível em: <<http://www.desenvolvimento.gov.br/sitio/interna/interna.php?area=5&menu=1161> > . 2009. Acesso em 30 de jun. 2009.

COFFMAN JR, 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**, vol. 9, pág. 49-62, 2006.

EMBRAER. **Embraer executive jets programs advance**. Disponível em: <
http://www.embraer.com.br/institucional/download/2_068-Com-VAE-Program_Updates_EBACE-I-08.pdf. > 2008. Acesso em 20 jul. 2009.

EMBRAER. **Programas Legacy 450 e Legacy 500 da Embraer avançam**. Disponível em:
<http://www.embraer.com.br/institucional/download/1_062-Prd-VAE-L450-500_Advance-P-09.pdf> 2009. Acesso em 20 jul. 2009.

HAX, A., CANDEA, D. **Production Inventory Management**. New Jersey: Prentice-Hall, 1984.

HOWE, D. **Aircraft loading and structural layout**. Reston, Virginia. American Institute of Aeronautics Inc (AIAA), pág. 391-422, 2004.

LAWLER, E. L., LENSTRA, L. K., RINNOOY, A. H. G., SHMOYS, D. B. Sequencing and Scheduling: Algorithms and Complexity. em: **Logistics of Production and Inventory**, Handbooks in Operations Research and Management Science, vol. 4, Graves, S. C., Rinnooy Kan, A. H. G., Zipikin, P. H. (eds), North-Holand, Elsevier, 455-522, 1993.

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, New York: Elsevier Butterworth-Heinemann, Oxford. pág. 211-232, 1999.

MORABITO, R., PUREZA, V., “Modelagem e Simulação”, em: **Metodologia de Pesquisa em Engenharia de Produção**, P. Cauchick (ed), Editora Campus/Elsevier, Rio de Janeiro, 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**, vol. 45, n. 1, pág. 1-8, 1999.

NAHMIAS, S. **Production and Operation Analysis**, 4th ed, Boston: McGraw-Hill, 2001.

NIU, M. C. Y. **Airframe structural design: practical design information and data on aircraft structures**. 1 ed. Hong Kong: Hong Kong Conmilit Press Ltda, 1988.

OLMOS, M., **Presidente da Airbus planeja parceria com a Embraer**. Disponível em: <http://www.defesanet.com.br/eads/br_10.htm>. 2008. Acesso em 13 jul. 2009.

OTTOBONI, J. **China endurece para comprar da Embraer**. Disponível em <<http://www.defesanet.com.br/emb1/cn.htm>>. 2007. 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**. 2nd ed., New Jersey, Prentice Hall, ,2008.

RODAMER, F. A., WHITE, K. P. A. Recent Survey of Production Scheduling. **IEE Transactions on Systems, Man and Cybernetics**, Vol. 18, pág. 841-851, 1988.

SANTOS, L. R. dos, SANTOS, R. H., ARENALES, M. N., RAGGI, L. A. Um Modelo para a programação de rotações de culturas. **Pesquisa Operacional**. Rio de Janeiro, v. 27, n. 3, 2007.

SANTOS, L. R. dos, MICHELON, P., ARENALES, M. N., SANTOS, R. H. S. Crop rotation scheduling with adjacency constraints. **Annals of Operations Research**. pág. 1-16, 2008.

SCHEMENNER, R. W. **Production, Operations Management – Concepts and Situation**. 4 ed., New York: MacMillan Publishing Company, pág. 674 – 705, 1990.

SUKHOI. **Komsomomolsk-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. **IV ALIO/EURO Workshop on Applied Combinatorial Optimization**. vol. 154, n. 5, abril, pág. 867-879, 2006.