



Otimização na programação de montagens na indústria aeronáutica

Assembly scheduling optimization in the aeronautical industry

Bruno Jensen Virginio da Silva¹
Reinaldo Morabito¹
Denise Sato Yamashita¹

Resumo: Este trabalho estuda um caso prático de programação da produção na indústria aeronáutica. Tarefas devem ser programadas em gabaritos de montagem com várias estações de trabalho adjacentes entre si, levando em consideração restrições especiais de processamento simultâneo destas tarefas (restrições de adjacência). Tais restrições surgem devido às limitações de espaço físico no gabarito. Este estudo é uma continuação do estudo anterior publicado em Silva, Morabito e Yanasse (2011), em que este problema foi inicialmente abordado do ponto de vista de minimizar o *makespan* para processar todas as tarefas. Aqui o problema é visto de forma mais abrangente no contexto de qualificação de força de trabalho com quatro fases, em que o objetivo é minimizar as necessidades de mão de obra para executar a programação em cada fase, sendo que cada fase de qualificação é representada por um modelo de otimização linear inteira. Estes modelos foram implementados em um *software* de otimização e vários experimentos computacionais foram realizados para verificar a aplicação da abordagem. Os resultados sugerem que existem ganhos de produtividade importantes que podem ser explorados utilizando os modelos desenvolvidos neste trabalho.

Palavras-chave: Programação da produção. Indústria aeronáutica. Montagem em gabaritos. Programação linear inteira. Restrições de adjacência.

abstract: *This paper presents a case study of production scheduling in the aeronautical industry. Tasks must be scheduled in assembly fixtures with several adjacent workstations, taking into account special constraints where simultaneous processing of the tasks is allowed (adjacency constraints). Such constraints arise due to a space limitation in the fixture. This paper is a continuation of a previous work published in Silva, Morabito and Yanasse (2011), where the objective of the problem was to minimize the makespan. Here, the problem is studied more broadly, where the workforce qualification is represented in four stages, with the objective of minimizing the total amount of workforce needed in each stage, and each stage is modeled as an integer linear programming problem. These models were implemented using an optimization software, and several computational experiments were performed in order to validate the proposed models. The results suggest that productivity gains can be achieved with the models developed in this work.*

keywords: *Scheduling. Aeronautical industry. Assembly in jigs. Linear programming. Adjacency constraints.*

1 Introdução

A indústria aeronáutica é caracterizada tipicamente pelo emprego de alta tecnologia no projeto, na produção e na operação dos seus produtos. A produção de aviões, particularmente a montagem de estruturas aeronáuticas, é predominantemente manual e dependente de mão de obra especializada e cara. Além disso, a indústria aeronáutica utiliza dispositivos fixos chamados *gabaritos* (NIU, 1988) para produção de partes e componentes montados, com

a finalidade de garantir que as condições de projeto do produto sejam mantidas durante as montagens dos aviões. Os gabaritos são específicos, ou seja, cada parte do avião possui gabaritos específicos para ser montada, são caros e precisam de um grande tempo para sua produção. A utilização adequada da mão de obra e dos gabaritos de produção é uma necessidade de qualquer fabricante de aviões para se manter competitivo no mercado.

¹ Departamento de Engenharia da Produção, Universidade Federal de São Carlos – UFSCar, Rod. Washington Luiz, Km 235, CP 767, CEP 13565-905, São Carlos, SP, Brasil, e-mail: brunojen@gmail.com; morabito@ufscar.br; dsyamashita@gmail.com

Recebido em 4/5/2011 — Aceito em 23/1/2012

Suporte financeiro: CAPES, CNPq, FAPESP.

Este trabalho estuda um caso de montagem na indústria aeronáutica, utilizando um caso prático presente em uma fabricante de aviões no Brasil. Operadores com diferentes graus de especialização devem montar peças componentes de um *subconjunto*, que é um pequeno conjunto de peças montadas, em gabaritos com várias estações de trabalho adjacentes. Conforme mencionado, em decorrência de restrições de espaço físico no gabarito, montagens em estações de trabalho adjacentes não podem acontecer simultaneamente, implicando em restrições de adjacência para a programação da produção no gabarito. Os objetivos deste estudo são obter a capacidade máxima de um gabarito, levando em conta os níveis de qualificação dos operadores e definir curvas de mão de obra (definidas na seção 4), que indicam quantas pessoas são necessárias na equipe de montagem para produzir aviões em diferentes cadências. Para isso, modelos matemáticos de programação de produção são utilizados para apoiar as decisões envolvidas.

Na indústria aeronáutica, à medida que aviões vão sendo produzidos, a quantidade necessária de horas-homem para se montar cada avião tende a diminuir, fenômeno esse conhecido na literatura como curva de aprendizagem (NAHMIA, 2008; ANZANELLO; FOGLIATTO, 2007). Diferente da literatura clássica, na qual o processo de aprendizagem pode ser modelado como uma função matemática (e.g., uma curva de aprendizagem exponencial decrescente), neste trabalho, ele é retratado genericamente como 4 fases, combinando aprendizagem e qualificação da força de trabalho com características distintas (descritas em mais detalhes na seção 3).

Convém salientar que o presente estudo é uma continuação do estudo anterior descrito em Silva, Morabito e Yanasse (2011), que tratou deste problema de programação em gabaritos e apresentou um modelo de programação matemática para encontrar a cadência máxima de produção de um gabarito apenas na fase de protótipo, que corresponde à primeira das 4 fases mencionadas acima (fase 1 da aprendizagem). Tal modelo foi inspirado em formulações matemáticas conhecidas da literatura de problemas de programação de *job shop*. No presente artigo, outros modelos de otimização são apresentados para representar as demais fases da produção seriada (fases 2, 3 e 4 da aprendizagem), inspirados em formulações matemáticas da literatura de gerenciamento e programação de projetos com tempo restrito. Estes modelos foram formulados como programas lineares inteiros e foram implementados e resolvidos com o aplicativo GAMS/CPLEX (BROOKE; KENDRICK; ROSENTHAL, 1998), utilizando dados reais coletados no setor de montagem de um fabricante brasileiro de aviões.

Não temos conhecimento de outros trabalhos na literatura que tenham estudado a programação de montagens de partes e componentes em gabaritos na indústria aeronáutica com o presente enfoque. Este trabalho está organizado da seguinte maneira: na seção 2, apresenta-se uma breve revisão de literatura relacionada; na seção 3, apresentam-se os modelos matemáticos desenvolvidos para abordar o problema; na seção 4, apresentam-se os resultados computacionais obtidos com estes modelos; e, finalmente, na seção 5, apresentam-se as conclusões deste estudo e algumas perspectivas para pesquisa futura.

2 Revisão de literatura

Algumas aplicações de técnicas de pesquisa operacional na indústria aeronáutica podem ser encontradas na literatura, por exemplo, em Dale (2001), Chikong, Chang e Lin (2006) e Abuabara e Morabito (2008, 2009). Neste estudo, foi realizada uma pesquisa bibliográfica procurando trabalhos que estudaram especificamente programação de montagens em gabaritos na indústria aeronáutica. Foram encontrados apenas dois trabalhos, Scott (1994) e Heike et al. (2001), que estudaram a programação de montagens estruturais nesta indústria, porém, estes trabalhos não trataram da programação de montagens em gabaritos. Aparentemente, a programação de montagens na indústria aeronáutica ainda é um tópico de pesquisa muito pouco explorado na literatura de programação (*scheduling*), especialmente a montagem em gabaritos com restrições de adjacência.

No entanto, a programação da produção com restrições de adjacência aparece na literatura de *scheduling*, em outras aplicações bem diferentes da indústria aeronáutica, e também com diversas nomenclaturas. Por exemplo, Weintraub et al. (2007) estudaram a programação da produção com restrições de adjacência na programação de colheitas de florestas; Santos et al. (2007, 2011) estudaram na programação de rotação de culturas; Gandham, Dawande e Prakash (2008) estudaram aplicações no contexto de telecomunicações; e Irani e Leung (1996, 2003) estudaram na programação de semáforos. Assim como o problema de programação em gabaritos, todos esses outros problemas têm em comum uma mesma restrição: se uma tarefa foi designada a um recurso, outras tarefas que utilizam outros recursos não podem ser processadas ao mesmo tempo devido às restrições de adjacência. Como nenhuma das abordagens propostas nestes trabalhos pode ser diretamente aplicada no problema de programação em gabaritos, optamos por investigar possíveis abordagens baseadas na literatura de gerenciamento e programação de projetos.

A programação da produção tem como objetivo distribuir as tarefas aos recursos ao longo do tempo,

de forma a atender todos os prazos acordados a um custo aceitável (MORTON; PENTICO, 1993). Já a programação de projetos é aplicada nos casos em que os produtos a serem produzidos são feitos apenas uma vez ou poucas vezes, com durações bem definidas no tempo (MORTON; PENTICO, 1993). Um dos principais problemas de programação de projetos é aquele em que, dados:

- uma quantidade limitada de recursos disponíveis;
- um conjunto de tarefas a serem executadas, com relações de precedência entre si; e
- a quantidade de recursos consumidos por cada tarefa.

consiste em alocar os recursos às tarefas respeitando todas as restrições, de maneira a otimizar um determinado objetivo.

Esse problema é conhecido como problema de programação de projetos com restrições de recursos (RCPSP – *resource constrained project scheduling problem*) (PINEDO, 2008). Um tipo especial de RCPSP é o caso em que o recurso limitado é o tempo, ou seja, dado um horizonte de tempo, o problema consiste em determinar quais recursos serão necessários para cumprir o projeto. Esse é chamado de problema de programação de projetos com restrição de tempo (TCPSP – *time constrained project scheduling problem*) (PINEDO, 2008). Estes dois problemas estão relacionados no conhecido *trade-off* entre prazo de tempo e recursos requeridos para execução do projeto. Quanto maior a quantidade de recursos, menor é o tempo necessário para executar o projeto, porém maior é o custo associado à quantidade de recursos utilizados no projeto. Quanto maior o tempo utilizado para executar o projeto, menor é o custo com os recursos utilizados para executar o projeto, porém maior é o custo de oportunidade. Brucker et al. (1999) apresentaram uma extensa revisão dos métodos utilizados para resolver esses problemas de programação de projetos.

Uma outra classe de problemas de programação de projetos também utilizada neste trabalho é o caso em que o RCPSP ou o TCPSP possui múltiplos modos de execução (MMRCPSP *multi-mode*

resource constrained project scheduling problem e MMTCPSP *multi-mode resource constrained project scheduling problem*, respectivamente). Nesse problema, tarefas podem ser executadas de formas diferentes, ou seja, consumindo quantidades e ou tipos de recursos diferentes em cada modo e, assim, apresentando durações diferentes dependendo do modo de execução escolhido. Esse tipo de problema foi estudado, por exemplo, em Yamashita e Morabito (2007a, b, 2009).

3 Modelagem matemática

Nesta seção, descreve-se cada uma das 4 fases de aprendizagem/qualificação dos operadores envolvidas no problema de programação em gabaritos na empresa aeronáutica estudada, e modela-se cada uma destas fases com base nas formulações conhecidas para os problemas RCPSP e TCPSP.

3.1 Apresentação do problema

A seguir, o problema de programação em gabaritos é resumidamente descrito; para mais detalhes deste problema o leitor pode consultar Silva (2010) e Silva, Morabito e Yanasse (2011). A montagem estudada neste trabalho refere-se à montagem de um subconjunto, formado por 8 peças. Cada peça é montada individualmente em um gabarito e, numa etapa posterior de montagem, são unidas, formando o subconjunto em estudo. Cada avião produzido utiliza dois desses subconjuntos. Cada uma das 8 peças recebe um nome, referente ao número da posição em que elas são montadas no subconjunto. A Figura 1 apresenta as peças que compõem o subconjunto e os seus respectivos números.

O gabarito utilizado para montar cada uma das peças do subconjunto é formado por 8 estações. Cada estação possui um número em que é montada a respectiva peça. Assim, a peça 1 é montada na estação 1, a peça 2 na estação 2, e assim sucessivamente. A restrição de adjacência acontece porque, devido às limitações de espaço físico, enquanto um operador está montando

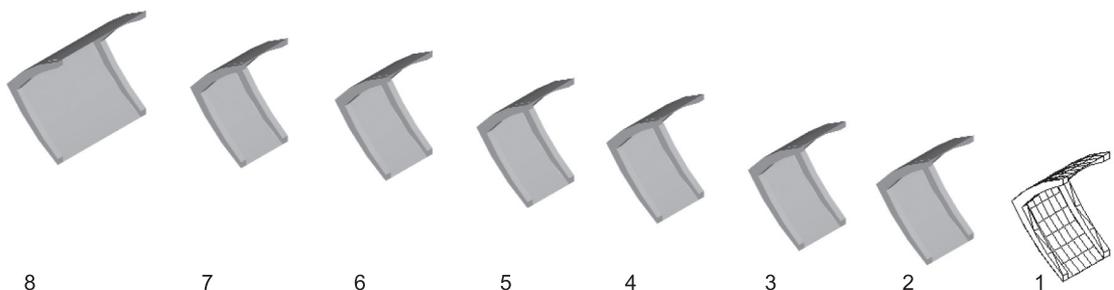


Figura 1. Subconjunto e as suas respectivas peças.

uma peça, por exemplo, na estação 3, as estações 2 e 4 não podem ser utilizadas para outras montagens, pois ficam bloqueadas. A Figura 2 apresenta o gabarito utilizado para montar as peças deste subconjunto.

Cada peça do subconjunto precisa executar pelo menos duas operações, uma no gabarito da Figura 2 e uma operação em bancadas ao lado do gabarito. Para montar um subconjunto inteiro, são necessárias 15 tarefas, com duas operações cada uma. Como cada avião precisa de dois subconjuntos, cada avião precisa de 30 tarefas, ou 60 operações, para estar completo nessa montagem. A Figura 3 apresenta a rede de precedência destas tarefas. As operações em branco são as operações no gabarito e as operações em cinza são as operações nas bancadas.

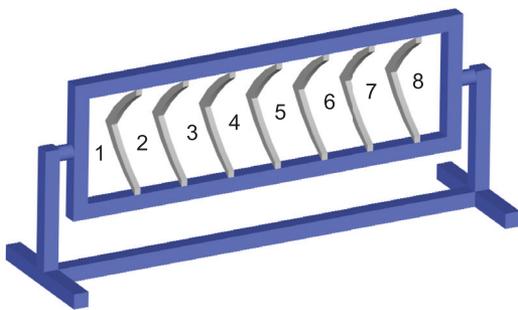


Figura 2. Gabarito múltiplo com restrições de adjacência utilizado na montagem do subconjunto.

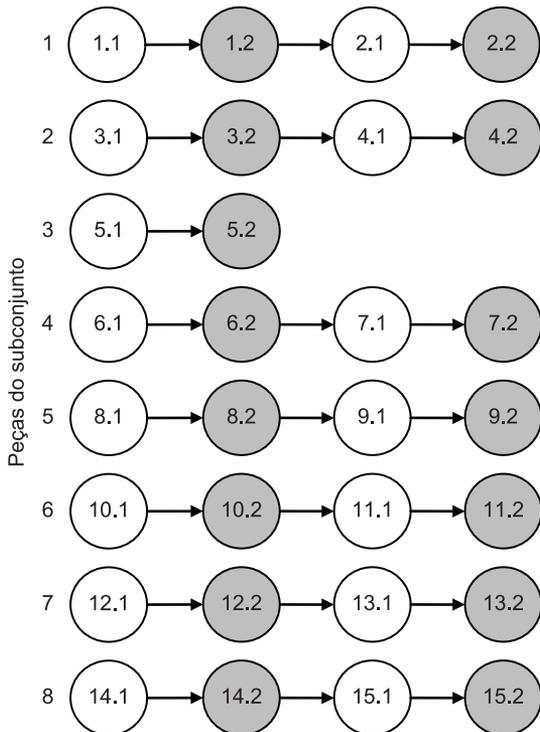


Figura 3. Rede de precedência da montagem de um subconjunto.

A montagem das peças do subconjunto foi dividida em 4 fases. A fase 1 é a fase de protótipo, em que os tempos de produção são desconhecidos ou estimados, e a rede de precedência das tarefas ainda não está bem definida. Nessa fase, a preocupação é determinar a capacidade máxima de produção do gabarito, com a finalidade de dimensionar a área de produção com a capacidade adequada para atender às cadências de montagem previstas. Nas fases 2, 3 e 4, o avião está em produção seriada. Em cada uma dessas fases, a equipe de montagem possui diferentes graus de aprendizagem, treinamento e polivalência. Na fase 2, a equipe é formada por dois grupos de operadores: os operadores especializados e os operadores não especializados. Os operadores especializados são capazes de executar apenas as operações de montagem dos gabaritos, enquanto a equipe não especializada é capaz de executar apenas as montagens de bancada. Na fase 3, a equipe especializada é capaz de executar as operações de montagem no gabarito e nas bancadas, enquanto a equipe não especializada é capaz de executar apenas as operações de montagem nas bancadas. Finalmente, na fase 4, a divisão entre operadores especializados e não especializados se desfaz, já que todos são capazes de executar qualquer operação de montagem.

3.2 Fase 1

Na fase 1, a produção em série não foi iniciada ainda. Os tempos de operação são estimados com base na produção dos protótipos. O objetivo nessa fase é encontrar qual a capacidade máxima de produção do gabarito, dada a rede de precedência e as durações conhecidas do projeto. Portanto, o objetivo é minimizar o tempo total para completar as tarefas, ou *makespan*. Conforme mencionado, um modelo matemático para representar a fase 1 foi apresentado recentemente em Silva, Morabito e Yanasse (2011) com base em formulações de programação da produção em *job shops*. O estudo desta fase não será objeto do presente trabalho, cujo foco está nas fases 2, 3 e 4. No entanto, convém observar que, uma vez encontrado o *makespan* na fase 1, podem ser utilizados qualquer um dos modelos das fases 2, 3 e 4 (apresentados adiante) para determinar qual a mão de obra mínima necessária para executar o programa com este *makespan*, ou seja, podem-se utilizar os modelos de programação de projetos com restrição de tempo para nivelar os recursos.

3.3 Fase 2

Na fase 2, inicia-se a produção em série dos aviões e os operadores são divididos em duas equipes. A equipe especializada realiza apenas as montagens de gabarito, enquanto a equipe não especializada executa apenas as montagens das bancadas. O objetivo é então

minimizar o custo ou a quantidade total de mão de obra utilizada para executar o programa de produção, dentro de um tempo determinado pelo tempo de ciclo da montagem do subconjunto.

Para apoiar as decisões de programação da fase 2, propõe-se a seguir um modelo matemático inspirado no clássico TCPSP (seção 2). Neste modelo, as restrições de adjacência são representadas por meio de um artifício, no qual cada posição do gabarito é composta de dois recursos fictícios. Assim, cada peça que será montada em uma estação de trabalho do gabarito utiliza os dois recursos dessa estação e mais um recurso de cada estação adjacente. Desta forma, quaisquer outras peças que são montadas nas estações adjacentes não têm os dois recursos fictícios disponíveis e, portanto, não podem ser montadas ao mesmo tempo.

A Figura 4 apresenta um exemplo ilustrativo. Nesse exemplo, é apresentado um gabarito com oito estações de trabalho, formadas por dois recursos fictícios. Considere que a tarefa 1, em azul, utiliza os dois recursos da estação 2 e um recurso das estações adjacentes 1 e 3. Dessa forma, as estações 1 e 3 estão bloqueadas enquanto a peça 2 é montada na estação 2 do gabarito. Considere também que uma outra tarefa, a tarefa 2, em verde, utiliza a estação 4 e, dessa forma, utiliza os dois recursos dessa estação, mais um recurso de cada estação adjacente, ou seja, as estações 3 e 5. Dessa forma, note que as estações 1, 3 e 5 ficam bloqueadas pelas tarefas 1 e 2 ao utilizarem as estações 2 e 4, devido às restrições de adjacência entre as estações.

O modelo matemático para a fase 2 pode ser formulado da seguinte maneira. Sejam os seguintes parâmetros:

- j tarefa $j = 1, \dots, J$
- t tempo $t = 1, \dots, T$
- k tipo de recurso, $k = 1, \dots, K$
- w tipo de mão de obra, $w = 1, \dots, W$
- r_j tempo em que a tarefa j está disponível, sendo que $1 \leq r_j \leq T$ e inteiro
- d_j tempo de entrega da tarefa j , sendo que $1 \leq d_j \leq T$ e inteiro
- rec_{jk} quantidade de recurso k necessária para executar a tarefa j

- c_k quantidade disponível do recurso k
- p_j duração da tarefa j
- mo_{jw} quantidade de mão de obra do tipo w utilizada para executar a tarefa j
- v_w custo unitário da mão de obra do tipo w

Além disso, define-se o seguinte conjunto de pares de tarefas (h, j) , sendo $h \neq j$.

- $H = \{(h, j) \mid \text{a tarefa } h \text{ precede a tarefa } j\}$.
- $G = \{j \mid \text{a tarefa } j \text{ é executada no gabarito}\}$

As variáveis de decisão do modelo da fase 2 são:

$$x_{jt} = \begin{cases} 1 & \text{se a tarefa } j \text{ termina no tempo } t \\ 0 & \text{caso contrário} \end{cases}$$

- a_w quantidade de mão de obra do tipo w a ser disponibilizada

Sendo que, neste caso particular da fase 2 (com $W = 2$), temos que:

- a_1 quantidade de mão de obra de gabarito a ser disponibilizada
- a_2 quantidade de mão de obra de bancada a ser disponibilizada

Então, o modelo matemático para a fase 2 é dado por:

$$\text{Min} \sum_{w=1}^W v_w a_w \tag{1}$$

$$\sum_{t=1}^T x_{jt} = 1 \text{ para } j = 1, \dots, J \tag{2}$$

$$\sum_{t=r_h+p_h}^{d_h} t \cdot x_{ht} \sum_{t=r_j+p_j}^{d_j} (t-p_j) \cdot x_{jt} \tag{3}$$

para todo $(h, j) \in H$

$$\sum_{j \in G} \sum_{b=t}^{t+p_j-1} rec_{jk} \cdot x_{jb} \leq c_k \text{ para todo } (k, t) \tag{4}$$

$$\sum_{j=1}^J \sum_{b=t}^{t+p_j-1} mo_{jw} \cdot x_{jb} \leq a_w \text{ para todo } (w, t) \tag{5}$$

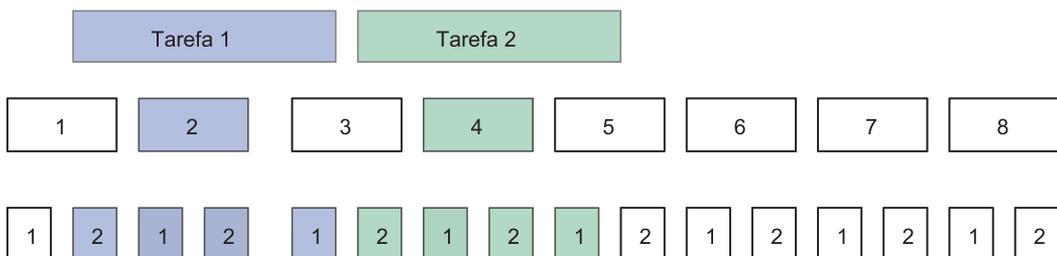


Figura 4. Representação dos recursos e utilização dos postos de trabalho do gabarito.

$$\begin{aligned}
 x_{jt} &= 0 & t = 1, \dots, r_j + p_j - 2 \\
 & & t = d_j + 1, \dots, T \\
 & \text{para } j = 1, \dots, J
 \end{aligned} \tag{6}$$

$$\{0,1\}; j = \dots, J; t = 1, \dots, T; a_w \in Z^+; w = 1, \dots, W \tag{7}$$

A função objetivo (1) minimiza o custo de mão de obra utilizada. Como no presente trabalho esse custo foi considerado igual para as duas equipes, a função objetivo (1) então minimiza a quantidade total de mão de obra utilizada. As restrições (2) garantem que cada tarefa seja alocada uma e somente uma vez em todo o horizonte de planejamento. As restrições (3) garantem que as relações de precedência entre as tarefas sejam respeitadas. As restrições (4) limitam a quantidade de recursos disponíveis, sendo que $t = b$ é o momento de término da tarefa j . As restrições (5) limitam a quantidade de mão de obra utilizada. As restrições (6) garantem que as datas de entrega e de chegada das tarefas sejam obedecidas. Finalmente, as restrições (7) referem-se ao domínio das variáveis de decisão.

3.4 Fase 3

Na fase 3, a equipe especializada pode realizar tanto as montagens de gabarito, quanto as montagens de bancada. Entretanto, a equipe não especializada pode montar apenas as operações de bancada. Deste modo, a divisão em duas equipes ainda se mantém. Para representar a fase 3, foi utilizado um modelo matemático baseado no MMTCPSP (seção 2), em que, para as tarefas de bancada, existem dois modos de execução. O primeiro modo utiliza uma mão de obra especializada, enquanto que o segundo modo utiliza uma mão de obra do grupo não especializado. O objetivo do problema continua sendo minimizar a quantidade total de mão de obra utilizada. Além dos parâmetros definidos anteriormente na seção, considere também os seguintes parâmetros adicionais para o modelo da fase 3 (alguns deles estão repetidos por conveniência):

- I modo de execução, $i = 1, \dots, I$
- rec_{jki} quantidade de recurso k necessária para executar a tarefa j no modo de operação i
- mog_{jw} quantidade de mão de obra do tipo w utilizada para executar a tarefa j no gabarito
- mob_{jwi} quantidade de mão de obra do tipo w utilizada para executar a tarefa j no modo de execução i na bancada

As variáveis de decisão do modelo da fase 3 são:

$$x_{jti} \begin{cases} 1 & \text{se a tarefa } j \text{ termina no tempo } t \text{ no modo} \\ 0 & \text{caso contrário} \end{cases}$$

- a_w quantidade de mão de obra do tipo w a ser disponibilizada
- Sendo que, neste caso da fase 3 ($W = 2$):
- a_1 quantidade de mão de obra de gabarito a ser disponibilizada
 - a_2 quantidade de mão de obra de bancada a ser disponibilizada

Por conveniência, definimos $J' = \{1, \dots, J\}$. O modelo matemático para a fase 3 é dado por:

$$\text{Min} \sum_{w=1}^W v_w a_w \tag{8}$$

$$\sum_{i=1}^I \sum_{t=1}^T x_{jti} = 1 \text{ para } j = 1, \dots, J \tag{9}$$

$$\sum_{i=1}^I \sum_{r_h+p_h}^{d_h} t \cdot x_{hti} \sum_{i=1}^I \sum_{r_j+p_j}^{d_j} (t - p_j) \cdot x_{jti} \tag{10}$$

para todo $(h, j) \in H$

$$\sum_{i=1}^I \sum_{j \in G} \sum_{b=t}^{t+p_j-1} rec_{jki} \cdot x_{jbi} \leq c_k \text{ para todo } (k, t) \tag{11}$$

$$\sum_{j \in G} \sum_{b=t}^{t+p_j-1} mog_{jw} x_{jbi} + \sum_{j \in J' \setminus G} \sum_{b=t}^{t+p_j-1} mob_{jwi} x_{jbi} \leq a_w \tag{12}$$

para $i = 1$ e $w = 1, t = 1, \dots, T$

$$\sum_{j \in J' \setminus G} \sum_{b=t}^{t+p_j-1} mob_{jwi} x_{jbi} \leq a_w \tag{13}$$

para $i = 2$ e $w = 2, t = 1, \dots, T$

$$\begin{aligned}
 x_{jti} &= 0 & t = 1, \dots, r_j + p_j - 2 \\
 & & t = d_j + 1, \dots, T \\
 & & j = 1, \dots, J \\
 & & i = 1, \dots, I
 \end{aligned} \tag{14}$$

$$x_{jti} = 0 \text{ } j \in G, i = 2, t = 1, \dots, T \tag{15}$$

$$\begin{aligned}
 x_{jti} &\in \{0,1\}; j = 1, \dots, J; t = 1, \dots, T; \\
 &i = 1, \dots, I; a_w \in Z^+; w = 1, \dots, W
 \end{aligned} \tag{16}$$

A função objetivo (8) minimiza o custo total de mão de obra utilizada. As restrições (9)-(16) têm as mesmas finalidades do modelo (1)-(7) da fase 2 apresentado na seção 3.3, porém adaptados para múltiplos modos de execução.

3.5 Fase 4

Na fase 4, todos os operadores disponíveis na área de montagem podem realizar qualquer tarefa; logo, a mão de obra é considerada apenas um único recurso.

Nessa fase, os operadores que eram especializados em bancadas já estão habilitados a trabalhar também no gabarito e então, a equipe é toda formada por operadores totalmente flexíveis. O objetivo é minimizar o tamanho da equipe, explorando a flexibilidade de todos os operadores. Note que a solução ótima do modelo matemático (1)-(7) da fase 2 com $W = 2$ (i.e., dois tipos de mão de obra) na seção 3.3, considerando o mesmo número de tarefas e duração de cada operação, a mesma quantidade de gabaritos e as mesmas datas de entrega e disponibilidade, é um limitante superior para a solução da fase 4. De fato, o problema da fase 4 também pode ser representado pelo modelo (1)-(7) da seção 3.3, com a diferença que na fase 4 não existem dois tipos de mão de obra, mas apenas um tipo (i. e., $W = 1$).

4 Resultados computacionais

Os problemas implementados neste trabalho utilizaram os dados do exemplo 3 recentemente apresentados em Silva, Morabito e Yanasse (2011). Esse exemplo contempla dados coletados na fase de protótipo e na fase de produção em série. Além da rede de precedência representada pelo conjunto de tarefas, é necessário conhecer a duração de cada tarefa, as relações de adjacência e identificar as tarefas que utilizam a mesma estação de trabalho no gabarito. Além disso, foi considerado que todas as tarefas estão disponíveis para serem iniciadas no momento $t = 0$ e todas têm o mesmo prazo de entrega, igual ao *makespan* do projeto. Os modelos matemáticos apresentados na seção 3 foram implementados em linguagem de programação matemática GAMS 23.0 (BROOKE; KENDRICKD; ROSENTHAL, 1998) e resolvidos com o CPLEX 11. O computador utilizado foi um Intel i7, com a opção do CPLEX para processamento em paralelo ativada, utilizando 4 núcleos (*threads*), e 12 GB de RAM. O tempo computacional máximo para o CPLEX resolver cada modelo foi limitado em 600 minutos (i.e., 10 horas).

Resultados computacionais da fase 1 para este exemplo foram apresentados em Silva, Morabito e Yanasse (2011). O tempo mínimo para completar a montagem no gabarito foi de 162 unidades de tempo (u.t.), sendo necessárias 13 pessoas. No presente trabalho, são apresentados os resultados computacionais das fases 2, 3 e 4 para este mesmo exemplo. Para cada uma destas fases, foram gerados 126 problemas-teste, variando-se apenas o tempo total para completar o projeto, ou seja, variando-se a cadência de produção de aviões por período de tempo entre a maior cadência possível e a menor cadência já praticada. A maior cadência possível corresponde ao *makespan* encontrado na solução da fase 1, ou seja, 162 u.t. A menor cadência já praticada pela empresa em questão foi 288 u.t. Todos os outros dados para os modelos das fases 2, 3 e 4, como durações das tarefas

e rede de precedência, são os mesmos do modelo da fase 1 em Silva, Morabito e Yanasse (2011).

4.1 Resultados computacionais da fase 2

Conforme mencionado, o modelo matemático (1)-(7) da fase 2 foi resolvido para vários problemas-teste, partindo-se do problema-teste com menor tempo de duração de projeto, 162 até 288 u.t., com a finalidade de avaliar o impacto da variação de cadência de produção na quantidade de mão de obra utilizada. O problema-teste com tempo de projeto de 288 u.t. possui 17.866 variáveis binárias e 8.295 equações, e foram necessários 49 minutos para o CPLEX encontrar a solução ótima, apresentada na Figura 5. A Figura 5 apresenta o gráfico de Gantt da solução do problema-teste com 288 u.t., apresentando separadamente as tarefas referentes às montagens de bancada e gabarito. Cada linha das tarefas de gabarito corresponde a uma estação de trabalho do gabarito e cada linha das tarefas de bancada corresponde a uma bancada diferente. Cada tarefa precisa de uma pessoa para ser executada, então, o número de tarefas sendo executadas em paralelo em qualquer momento do gráfico representa a quantidade de operadores executando montagens. Perceba que as tarefas de montagem no gabarito apresentam um padrão com vários espaços em branco, o que significa que aquela estação de trabalho do gabarito está ociosa, seja por restrição de adjacência, precedência ou disponibilidade de mão de obra.

A Figura 6 apresenta a variação da quantidade de mão de obra utilizada durante o programa, em cada uma das duas equipes de montagem. Note que são necessárias 3 pessoas na equipe especializada e 5 pessoas na equipe não especializada, totalizando 8 pessoas. A Figura 7 apresenta a variação da quantidade total de mão de obra durante a execução do programa. Perceba na Figura 7 que, embora sejam necessárias 8 pessoas no total, em nenhum momento as 8 pessoas trabalham simultaneamente. Em apenas alguns poucos momentos, são necessários 7 operadores trabalhando ao mesmo tempo, como, por exemplo, nos momentos 83 u.t. e 124 u.t.

A Figura 8 apresenta o tamanho ótimo da equipe de montagem em função do tempo para completar o projeto. Perceba que os pontos apresentam um padrão de degraus, ou seja, aceitam-se somente quantidades inteiras de pessoas nas equipes. Perceba também que, para uma certa quantidade de pessoas na equipe, existem vários tempos de projeto ou cadências de produção que são possíveis de serem atendidos. Assim, pode-se ter uma medida da capacidade de reação de uma equipe, aumentando-se a sua velocidade de produção, sem alterar a quantidade total de operadores. Esta curva da Figura 8 é chamada de curva de mão de

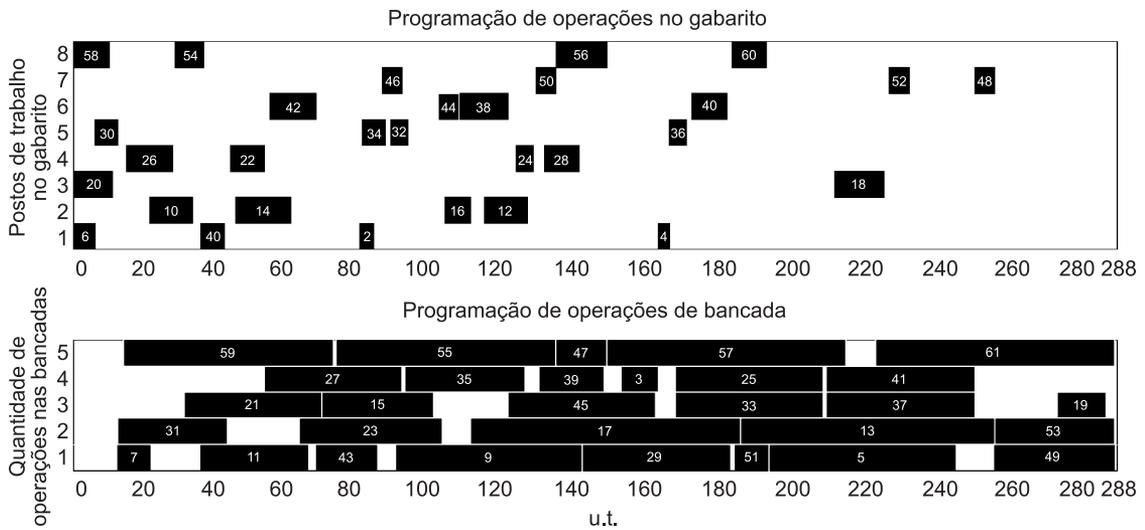


Figura 5. Gráfico de Gantt com o programa de produção na fase 2 para o *makespan* 288 u.t.

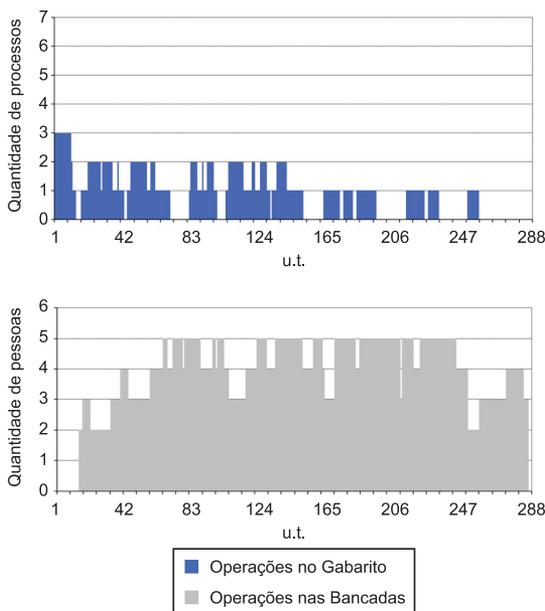


Figura 6. Variação do número de operadores no gabarito e na bancada para a fase 2 do subconjunto para o *makespan* 288 u.t.

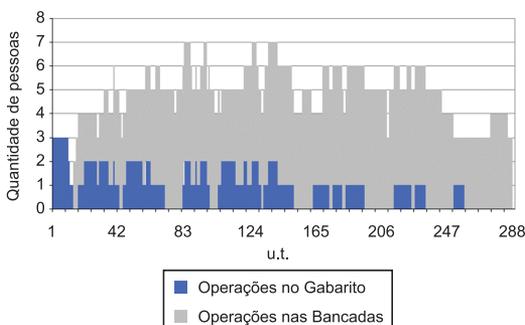


Figura 7. Variação do número de operadores na fase 2 para o *makespan* 288 u.t.

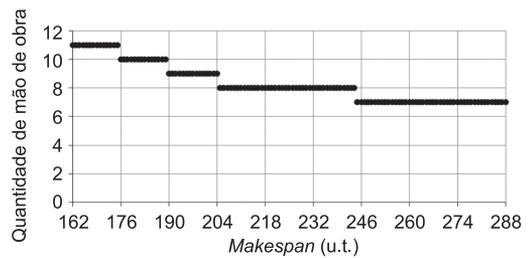


Figura 8. Tamanho ótimo das equipes de montagem em função do *makespan* para a fase 2.

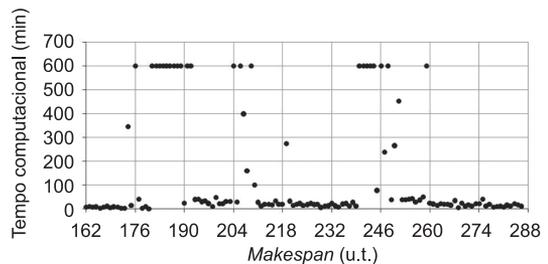


Figura 9. Tempo computacional para cada um dos *makespan* na fase 2.

obra. Cada degrau da curva é um tamanho de equipe que atende a um intervalo de *makespan*.

Considere, por exemplo, a equipe de tamanho 9. Ela consegue montar aviões com *makespan* entre 190 e 209 u.t. Assim, por exemplo, se o *makespan* do projeto é 190 u.t., qualquer pequena variação que necessite de um *makespan* menor vai precisar de uma pessoa a mais, ou seja, 10 pessoas, para poder finalizar a montagem dentro do prazo. Se, ao contrário, o *makespan* precisar aumentar, basta utilizar uma solução com o *makespan* novo e não é preciso alterar o tamanho da equipe. Caso o *makespan* seja de 209 u.t., uma diminuição no *makespan* é tolerável, pois ele

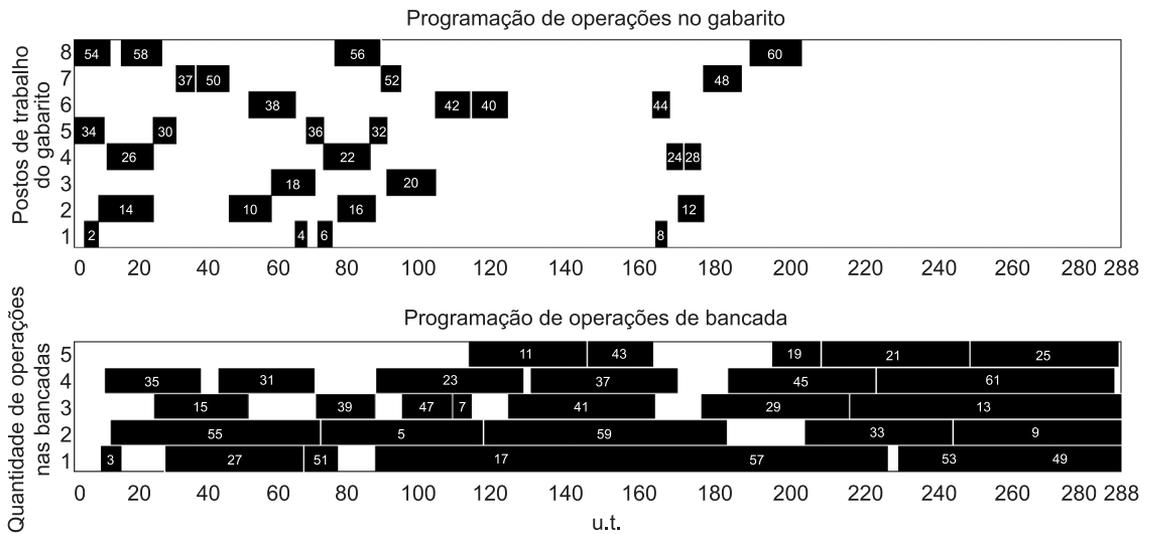


Figura 10. Gráfico de Gantt com o programa de produção na fase 3 para *makespan* 288 u.t.

pode diminuir até 190 u.t. sem alterar o tamanho da equipe. Nesse caso, se o *makespan* precisar ser apenas um pouco maior, a equipe terá uma pessoa sobrando. Dessa forma, a curva de mão de obra permite ao responsável pela área entender melhor qual a capacidade de reação da equipe, caso seja necessário algum plano de contingência.

A Figura 9 apresenta a variação do tempo computacional para encontrar a solução dos problemas-teste estudados. Perceba que vários problemas-teste alcançaram o tempo computacional máximo, 600 minutos, conseqüentemente, o CPLEX não foi capaz de provar a otimalidade da solução encontrada dentro do tempo estipulado. Ainda assim, em casos específicos, é possível deduzir o valor da solução ótima por meio de comparação com limitantes superiores e inferiores. Para isto, basta conhecer as soluções ótimas dos problemas-teste com *makespan* imediatamente maior e menor. Se esses forem iguais, então a solução do problema-teste, do qual não se conhece a otimalidade, só poderá ser a mesma dos dois problemas-teste vizinhos.

4.2 Resultados computacionais da fase 3

Nessa fase, a equipe especializada pode fazer qualquer montagem, enquanto a equipe não especializada realiza apenas as montagens de bancada. O programa de produção do problema-teste com *makespan* 288 u.t. é apresentado na Figura 10. A Figura 11 apresenta a variação da quantidade de mão de obra executando as operações de bancada e gabarito. A Figura 12 apresenta a variação na quantidade total de mão de obra. Note que são necessárias 5 pessoas no total para executar o programa, com 3 pessoas

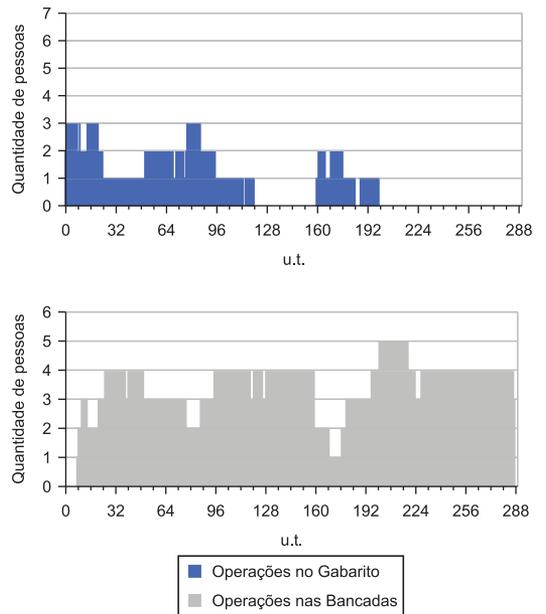


Figura 11. Variação do número de operadores no gabarito e na bancada para a fase 3 para o *makespan* 288 u.t.

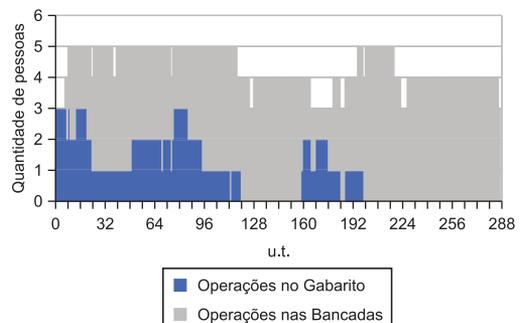


Figura 12. Variação do número de operadores no exemplo 3 na fase 3 para o *makespan* 288 u.t.

na equipe especializada e 2 pessoas na equipe não especializada.

A Figura 13 apresenta a variação da quantidade de mão de obra em função do *makespan*. Como nessa fase é utilizado o modelo de programação de projetos com múltiplos modos de execução, é esperado um desempenho computacional pior em relação ao modelo da fase 2, que utiliza o modelo monomodo. Por isso, vários problemas-teste da fase 3 apresentam soluções piores do que os respectivos problemas-teste da fase 2. A Figura 14 apresenta o tempo computacional

necessário para encontrar as soluções apresentadas na Figura 14. Perceba que muitos problemas-teste alcançam o tempo computacional máximo, de 600 minutos. Isso significa que as soluções encontradas com esse tempo computacional não têm otimalidade comprovada.

4.3 Resultados computacionais da fase 4

Na fase 4, não existe mais divisão em duas equipes, ou seja, considera-se que a equipe de montagem é capaz de realizar qualquer montagem. A Figura 15 apresenta

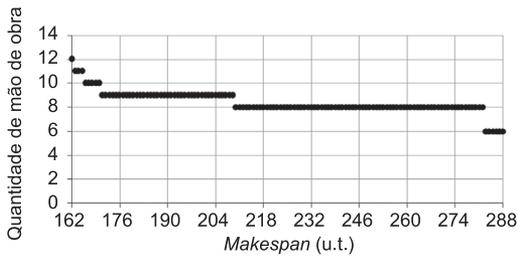


Figura 13. Tamanho das equipes de montagem em função do *makespan* para a fase 3.

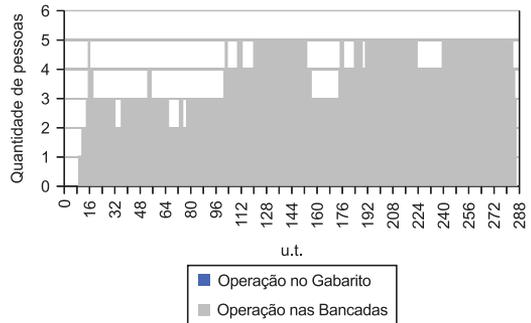
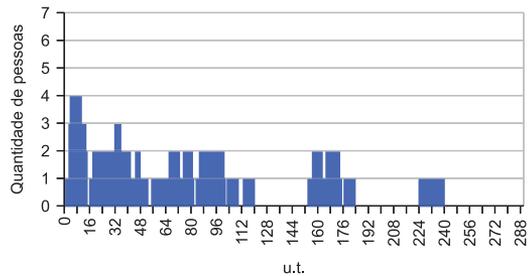


Figura 16. Variação do número de operadores no gabarito e na bancada para a fase 4 para o *makespan* 288 u.t.

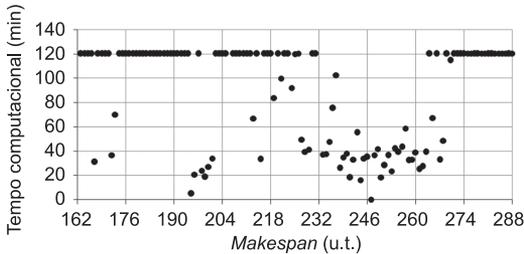


Figura 14. Tempo computacional para cada um dos *makespan* na fase 3.

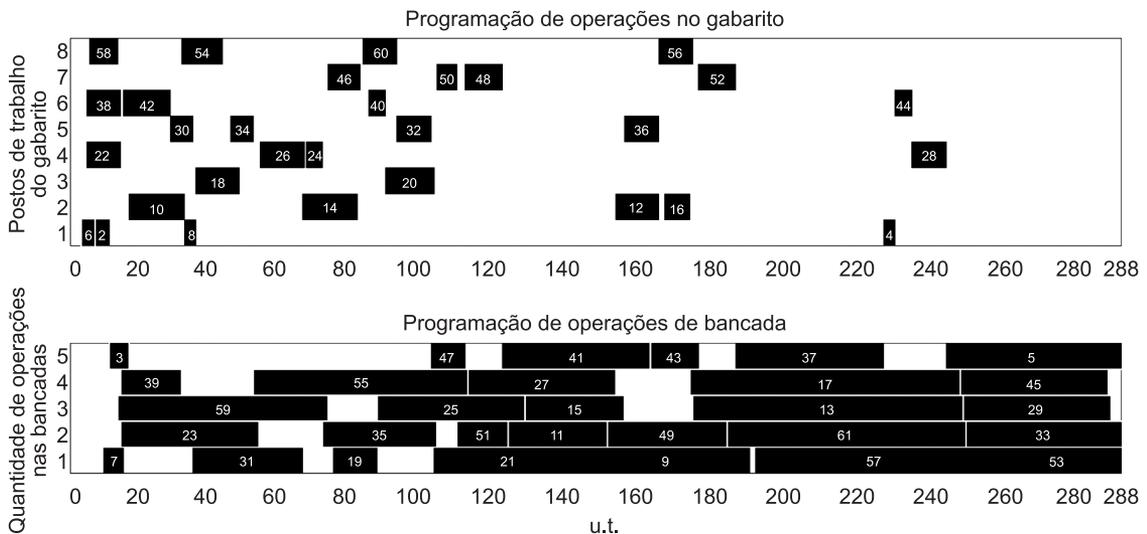


Figura 15. Gráfico de Gantt com o programa de produção na fase 4 para o *makespan* 288 u.t.

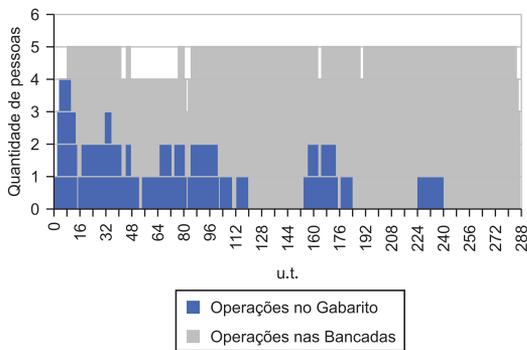


Figura 17. Variação do número de operadores na fase 4 para o *makespan* 288 u.t.

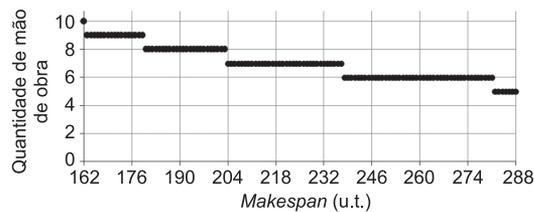


Figura 18. Tamanho ótimo das equipes de montagem em função do *makespan*.

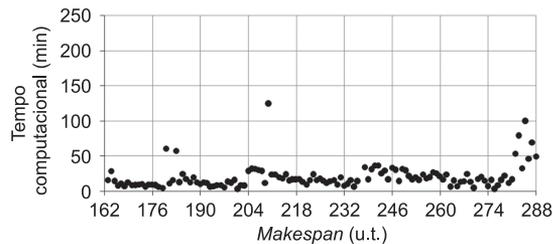


Figura 19. Tempo computacional para cada um dos *makespan* na fase 4.

o programa de produção para o problema-teste com *makespan* de 288 u.t. A Figura 16 apresenta a variação da quantidade de operações de gabarito e de bancada ao longo do programa. A Figura 17 apresenta a variação da quantidade de operadores ao longo do programa. Note que nesse caso são necessários no máximo 5 operadores para executar o programa.

A Figura 18 apresenta a variação da quantidade de mão de obra em função do *makespan*. A Figura 19 apresenta o tempo computacional para encontrar as soluções apresentadas na Figura 18. Como essa fase utiliza o modelo de programação de projetos com restrição de tempo, os tempos computacionais são menores que no caso da fase 3.

5 Conclusões

Os modelos matemáticos propostos neste estudo para representar o problema de montagem em gabaritos com restrições de adjacência mostraram que é possível utilizar abordagens baseadas em programação de

projetos para resolver esse problema e que os tempos computacionais, embora em alguns casos sejam da ordem de várias horas, são aceitáveis para o contexto de decisões envolvidas na prática de uma empresa aeronáutica. Os experimentos computacionais também mostraram que existe potencial de ganho de produtividade ao utilizar estes modelos para otimizar a programação da montagem. Em comparação com a prática da empresa, os resultados indicam que é possível reduzir até 30% a mão de obra utilizada, sem comprometer, significativamente, o atendimento aos prazos estabelecidos.

As chamadas curvas de mão de obra são também um resultado interessante deste estudo. Com elas, é possível avaliar com precisão e rapidez qual o impacto no efetivo da montagem estudada para diferentes variações de cadência de produção. Também é possível montar planos de contingência com certa facilidade e rapidez, para saber qual é a quantidade de mão de obra necessária para tratar uma situação imprevista na programação da produção. Algumas perspectivas interessantes para pesquisa futura a partir do presente estudo seriam a consideração de incertezas nos parâmetros dos modelos utilizando, por exemplo, técnicas de programação estocástica e otimização robusta.

Agradecimentos

Os autores agradecem aos dois revisores anônimos e ao Prof. Horácio H. Yanasse os seus úteis comentários e sugestões. Os autores agradecem também à empresa que colaborou com esta pesquisa e permitiu a coleta de dados necessária ao estudo. Essa pesquisa contou com o apoio financeiro da CAPES, CNPq e FAPESP.

Referências

- ABUABARA, A.; MORABITO, R. Modelos de programação inteira mista para o planejamento do corte unidimensional de tubos metálicos na indústria aeronáutica agrícola. **Gestão & Produção**, v. 15, n. 3, p. 605-617, 2008.
- ABUABARA, A.; MORABITO, R. Cutting optimization of structural tubes to build agricultural light aircrafts. **Annals of Operations Research**, v. 169, n. 1, p. 149-165, 2009. <http://dx.doi.org/10.1007/s10479-008-0438-7>
- ANZANELLO, M. J.; FOGLIATTO, F. S. Curvas de aprendizado: estado da arte e perspectivas de pesquisa. **Gestão & Produção**, v. 14, n. 1, p. 109-123, 2007. <http://dx.doi.org/10.1590/S0104-530X2007000100010>
- BROOKE, A.; KENDRICK, M. A. R. R.; ROSENTHAL, R. **GAMS: A User's Guide**. Washington: GAMS Development Corporation, 1998. 276 p.
- BRUCKER, P. et al. Resource-constrained project scheduling: notation, classification, models, and methods. **European Journal of Operational Research**, v. 112, p. 3-41, 1999. [http://dx.doi.org/10.1016/S0377-2217\(98\)00204-5](http://dx.doi.org/10.1016/S0377-2217(98)00204-5)
- CHIKONG, H.; CHANG, H.-M.; LIN, C.-H. Simulation and analysis for layout of aerospace sheet metal process with

- lean concept. **Materials Science Forum**, v. 505-507, p. 907-912, 2006. <http://dx.doi.org/10.4028/www.scientific.net/MSF.505-507.907>
- DALE, R. Offline programming and simulation help Boeing use giant automated riveter on C-17 aircraft. **Industrial Robot: An International Journal**, v. 28, n. 6, p. 478-482, 2001.
- 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)
- GANDHAM, S.; DAWANDE, M.; PRAKASH, R. Link scheduling in wireless sensor networks: distributed edge-coloring revisited. **Journal of Parallel and Distributed Computing**, v. 68, p. 1122-1134, 2008. <http://dx.doi.org/10.1016/j.jpdc.2007.12.006>
- IRANI, S.; LEUNG, V. Scheduling with conflicts and applications to traffic signal control. In: ACM SYMPOSIUM ON DISCRETE ALGORITHMS, 1996, Atlanta. **Proceedings...** ACM, 1996.
- IRANI, S.; LEUNG, V. Scheduling with conflicts on bipartite and interval graphs. **Journal of Scheduling**, v. 6, p. 287-307, 2003. <http://dx.doi.org/10.1023/A:1022908509269>
- MORTON, T. E.; PENTICO, D. W. **Heuristic scheduling systems: with applications to production systems and project management**. New York: John Wiley & Sons, 1993.
- NAHMIAS, S. **Production and operations analysis**. 4th ed. McGraw-Hill: New York, 2008.
- NIU, M. C. Y. **Airframe structural design: practical design information and data on aircraft structures**. Hong Kong: Hong Kong Conmilit Press Ltda, 1988.
- PINEDO, M. L. **Scheduling: theory, algorithms and systems**. 3rd ed. New-Jersey: Prentice-Hall, 2008.
- SANTOS, L. R. et al. Um Modelo para a programação de rotações de culturas. **Pesquisa Operacional**, v. 27, n. 3, p. 535-547, 2007. <http://dx.doi.org/10.1590/S0101-74382007000300007>
- SANTOS, L. M. R. et al. Crop rotation scheduling with adjacency constraints. **Annals of Operations Research**, v. 190, n. 1, p. 165-180, 2011. <http://dx.doi.org/10.1007/s10479-008-0478-z>
- SCOTT, H. A. Modeling aircraft assembly operations. In: WINTER SIMULATION CONFERENCE, 1994, San Diego. **Proceedings...** San Diego: Society for Computer Simulation International, 1994.
- SILVA, B. J. V. **Programação de montagens em gabaritos com restrições de adjacência na indústria aeronáutica**. 2010. 118 f. Dissertação (Mestrado em Engenharia de Produção)-Universidade Federal de São Carlos, São Carlos, 2010.
- SILVA, B. J. V.; MORABITO, R.; YANASSE, H. H. Programação de montagens em gabaritos com restrições de adjacência na indústria aeronáutica. **Produção**, v. 21, n. 4, p. 696-707, 2011. <http://dx.doi.org/10.1590/S0103-65132011005000055>
- WEINTRAUB, A. et al. (Eds.). **Handbook of operations research in natural resources**. New York: Springer, 2007. (International series in operations research & management science, v. 99).
- YAMASHITA, D. S.; MORABITO, R. Um algoritmo exato para o problema de programação de projetos com custo de disponibilidade de recursos e múltiplos modos. **Pesquisa Operacional**, v. 27, n. 1, p. 27-49, 2007a.
- YAMASHITA, D. S.; MORABITO, R. Um algoritmo branch-and-bound para o problema de programação de projetos com custo de disponibilidade de recursos e múltiplos modos. **Gestão & Produção**, v. 14, n. 13, p. 545-555, 2007b.
- YAMASHITA, D. S.; MORABITO, R. A note on time/cost tradeoff curve generation for project scheduling with multi-mode resource availability costs. **International Journal of Operational Research**, v. 5, n. 4, p. 429-444, 2009. <http://dx.doi.org/10.1504/IJOR.2009.025702>