

**UNIVERSIDADE FEDERAL DE SÃO CARLOS  
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA  
PROGRAMA DE PÓS-GRADUAÇÃO EM  
ENGENHARIA DE PRODUÇÃO**

**PROGRAMAÇÃO DE MONTAGENS EM  
GABARITOS COM RESTRIÇÕES DE  
ADJACÊNCIA NA INDÚSTRIA AERONÁUTICA**

Bruno Jensen Virginio da Silva

**SÃO CARLOS  
2010**

Programação de Montagens em Gabaritos com Restrições  
de Adjacência na Indústria Aeronáutica

**UNIVERSIDADE FEDERAL DE SÃO CARLOS  
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA  
PROGRAMA DE PÓS-GRADUAÇÃO EM  
ENGENHARIA DE PRODUÇÃO**

**PROGRAMAÇÃO DE MONTAGENS EM  
GABARITOS COM RESTRIÇÕES DE  
ADJACÊNCIA NA INDÚSTRIA AERONÁUTICA**

Bruno Jensen Virginio da Silva

**Dissertação de Mestrado  
apresentada ao Programa de  
Pós-Graduação em Engenharia  
de Produção da Universidade  
Federal de São Carlos como  
parte dos requisitos para a  
obtenção do Título de Mestre  
em Engenharia de Produção  
Orientador: Prof. Reinaldo Morabito**

**SÃO CARLOS  
2010**

**Ficha catalográfica elaborada pelo DePT da  
Biblioteca Comunitária da UFSCar**

S586pm

Silva, Bruno Jensen Virginio da.

Programação de montagens em gabaritos com restrições de adjacência na indústria aeronáutica / Bruno Jensen Virginio da Silva. -- São Carlos : UFSCar, 2011. 118 f.

Dissertação (Mestrado) -- Universidade Federal de São Carlos, 2010.

1. Programação da produção. 2. Gabaritos de montagem. 3. Indústria aeronáutica. 4. Restrições de adjacência. I. Título.

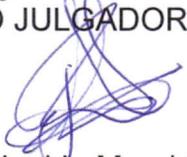
CDD: 658.53 (20ª)



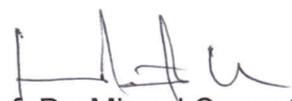
## FOLHA DE APROVAÇÃO

Aluno(a): Bruno Jensen Virginio da Silva

DISSERTAÇÃO DE MESTRADO DEFENDIDA E APROVADA EM 03/12/2010 PELA  
COMISSÃO JULGADORA:

  
Prof. Dr. Reinaldo Morabito Neto  
Orientador(a) PPGE/UFSCar

  
Prof. Dr. Horacio Hideki Yanasse  
LAC/INPE

  
Prof. Dr. Miguel Cezar Santoro  
POLI/USP

  
Profª Drª Denise Sato Yamashita

  
Prof. Dr. Roberto Antonio Martins  
Coordenador do PPGE

*Aos meus pais, Neuza e José  
que me prepararam para realizar meus sonhos  
muito antes de eu sequer sonhar.*

*“Ahhhhhhhhhhh!  
Ahhhhhhhhhhh!  
Finalmente! Eu não acredito!  
Ahhhhhhhhhhh!  
Putaquepariu!  
Ahhhhhhhhhhh!”*

Airton Senna, quando ganhou em Interlagos pela primeira vez, 1991.

# Agradecimentos

Agradeço aos meus pais, Neuza e José, pelo apoio sob as mais diversas formas possíveis.

Agradeço ao meu orientador, Reinaldo Morabito, que desde a graduação me guiou pela ciência e pela minha profissão, pela paciência e confiança.

Agradeço à empresa que permitiu esse estudo.

Agradeço ao Horácio Yanasse e à Denise Yamashita, pelos úteis comentários e idéias criativas para o trabalho, especialmente para os modelos.

Agradeço aos bons amigos, que ainda não entendem o que eu estudo, mas sempre me motivaram ir em frente.

Agradeço a todas as pessoas que se dedicaram e se dedicam à ciência, especialmente à Pesquisa Operacional. Sem essa dedicação coletiva, esse trabalho nunca existiria.

Este trabalho teve o apoio da CAPES, que concedeu bolsa durante a realização da pesquisa.

## Resumo

Este trabalho aborda um problema de programação da produção que ocorre 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 a quantidade total de mão de obra necessária para realizar a montagem em questão e determinar a capacidade de produção do gabarito, ou seja, minimizar o *makespan* (tempo total para executar as tarefas). 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. A montagem e as equipes que nela trabalham estão em melhoria contínua dos processos de montagem, conforme a produção acumulada aumenta e a curva de aprendizagem é percorrida. A curva de aprendizagem foi dividida em quatro fases com características específicas em cada uma. Propõem-se modelos de programação linear inteira mista para representar o problema de programação em cada uma das fases, com base em estudos de casos práticos de programação de gabaritos de montagem de uma empresa aeronáutica. Estes modelos são resolvidos utilizando-se uma linguagem de modelagem e um *software* de otimização. As soluções encontradas são analisadas e comparadas e com base nelas, avalia-se como a curva de aprendizagem afeta a produção de aviões ao longo do tempo e demonstra-se que existe potencial de melhoria no uso dos recursos de montagem, utilizando programação da produção e nivelamento dos recursos.

**Palavras chave:** programação da produção, gabaritos de montagem, restrições de adjacência, indústria aeronáutica.

## Abstract

This work deals with a production scheduling problem that 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 be performed in these workstations in order to minimize the quantity of manpower needed to do the assembly and determine the production capacity of the assembly fixture, which means minimizing the makespan (the total time needed to finish the tasks). However, in addition to the usual constraints such as due dates and precedence among tasks, there are also constraints that prevent two tasks to be performed at the same time in two adjacent workstations on the assembly fixture. The assembly and the teams who work there are in continuous improvement of assembling processes, according to the cumulative production increases and the learning curve is traversed. The learning curve was divided in four stages with specific characteristics in each. Mixed integer linear programming models are proposed to represent the production scheduling problem of each stage, based on practical cases studies of assembly fixtures scheduling in an aeronautics company. Those models are solved using a modeling language and optimization software. The solutions are analyzed and compared and based on then, it is evaluated how the learning curve affects the airplane production over time and it is shown, in the results, that there exists potential of improvement in the use of assembling resources, using production scheduling and levelling the resources.

**Key-words:** production scheduling, assembly fixtures, adjacent constraints, aeronautics industry.

## Lista de Ilustrações

Figura 1:	Vôo do 14 Bis, Campo Bagatelle, Paris (NASA, 2006).....	6
Figura 2:	Exportação da Embraer em US\$ bilhões nos últimos 10 anos (BRASIL, 2009).....	7
Figura 3:	Principais partes que compõem o avião (NASA, 2008).....	9
Figura 4:	Fuselagem seção transversal (BOMBARDIER AEROSPACE, 2009).....	10
Figura 5:	Painéis e estruturas de cavernas e vigas omega (STANFORD, 2009).....	10
Figura 6:	Exemplos de asas e componentes (U.S.A., 2008). ....	11
Figura 7:	Empenagem (U.S.A., 2003).....	11
Figura 8:	Montador e o respectivo gabarito de montagem de um subconjunto (AERO, 2009).....	14
Figura 9:	Máquina CNC que faz a rebitagem de painéis (BRÖTJE, 2009).....	15
Figura 10:	Montagem estrutural de um Boeing 787 (BOEING, 2009) .....	16
Figura 11:	Etapas de produção dos aviões .....	17
Figura 12:	Gabarito dedicado para montagem de um estabilizador horizontal (PILATUS, 2009). ....	18
Figura 13:	Gabarito flexível, para montagem de painéis (EVOLUTION MAGAZINE, 2006). ....	19
Figura 14:	Fluxo de trabalho de um <i>job shop</i> (traduzido de BAKER, 1974). ....	25
Figura 15:	Exemplo de programa de produção .....	26
Figura 16:	Formas de representação da rede de relacionamento das tarefas de um projeto (PINEDO, 2005). ....	29
Figura 17:	(a) Representação informal das atividades nos arcos (b) Representação formal das atividades nos arcos (MORTON; PENTICO, 1993). ....	30
Figura 18:	Tarefas que bloqueiam os recursos e um Gantt de um programa ilustrativo.....	37
Figura 19:	Tarefas que utilizam os recursos e um Gantt de um programa ilustrativo.....	37
Figura 20:	Grafo de conflito dos exemplos apresentados na figura 19 e figura 20 .	38
Figura 21:	Floresta dividida em áreas de manejo e o respectivo grafo representando as relações de adjacência das áreas de manejo (GOYCOOLEA et al., 2005).....	39
Figura 22:	Plantação de hortaliças e o respectivo grafo de adjacência (SANTOS et al., 2008).....	41
Figura 23:	(a) As antenas A e C não estão na mesma área, mas conflitam quando transmitem; (b) Rede <i>wireless</i> com os servidores AP1 e AP2 escondidos, mas com clientes na mesma área (RAY et al., 2005). ....	42
Figura 24:	Exemplo de uma rede com 100 nós (SHEU et al., 2008).....	43
Figura 25:	Representação de um cruzamento comandado por um semáforo e o respectivo grafo de conflito (IRANI; LEUNG, 2003).....	45
Figura 26:	Exemplo de programação de recursos adjacentes (DUIN; SLUIS, 2006) .....	46
Figura 27:	Curva de aprendizagem de 80% .....	49
Figura 28:	Subconjunto e as suas respectivas peças .....	52
Figura 29:	Gabarito múltiplo com restrições de adjacência utilizado na montagem do subconjunto.....	53

Figura 30:	Rede de relações das tarefas de montagem do subconjunto.....	54
Figura 31:	Grafo de adjacência das estações de trabalho.....	54
Figura 32:	Grafo das relações de adjacência das tarefas de dois subconjuntos inteiros .....	55
Figura 33:	Segundo subconjunto e as suas respectivas peças .....	55
Figura 34:	Gabarito múltiplo com restrições de adjacência utilizado na montagem do subconjunto 2.....	56
Figura 35:	Rede de relações das tarefas de montagem .....	57
Figura 36:	Grafo de adjacência das estações de trabalho.....	58
Figura 37:	Grafo das relações de adjacência das tarefas do segundo subconjuntos inteiros .....	58
Figura 38:	As duas estratégias de programação da produção dos aviões .....	60
Figura 39:	Fases da montagem enquanto percorre a curva de aprendizagem.....	62
Figura 40:	Representação dos recursos e utilização dos postos de trabalho do gabarito .....	65
Figura 41:	Rede de precedência das tarefas de montagem do subconjunto no exemplo 1 .....	73
Figura 42:	Gráfico de Gantt com o programa de produção do exemplo 1 na fase 1 do subconjunto com o modelo (42)-(51) .....	74
Figura 43:	Variação do número o de operadores no gabarito e na bancada para a fase 1 do exemplo 1 utilizando o modelo (42)-(51) .....	75
Figura 44:	Variação do número de operações para o exemplo 1 com o modelo (42)- (51).....	76
Figura 45:	Gráfico de Gantt com o programa de produção do exemplo 1 na fase 1 do modelo (52)-(58) .....	77
Figura 46:	Variação do número de operadores no gabarito e na bancada para a fase 1 do exemplo 1 do modelo (52)-(58).....	78
Figura 47:	Variação do número de operadores para o exemplo 1 com o modelo (52)- (58).....	79
Figura 48:	Rede de precedência das tarefas de montagem do subconjunto no exemplo 2 .....	80
Figura 49:	Gráfico de Gantt com o programa de produção do exemplo 1 na fase 1 do modelo (42)-(51) .....	81
Figura 50:	Variação do número de operadores no gabarito e na bancada para a fase 1 do exemplo 1 do modelo (42)-(51).....	82
Figura 51:	Variação do número de operadores para o exemplo 1 com o modelo (42)- (51).....	83
Figura 52:	Gráfico de Gantt com o programa de produção do exemplo 2 na fase 1 do modelo (52)-(58) .....	84
Figura 53:	Variação do número de operadores no gabarito e na bancada para a fase 1 do exemplo 2 com o modelo (52)-(58).....	85
Figura 54:	Variação do número de operadores no exemplo 2 .....	86
Figura 55:	Gráfico de Gantt com o programa de produção do exemplo 2 na fase 2	87
Figura 56:	Variação do número de operadores no gabarito e na bancada para a fase 2 do subconjunto.....	88
Figura 57:	Variação do número de operadores na fase 2 .....	89
Figura 58:	Tamanho ótimo das equipes de montagem em função do <i>makespan</i> para a fase 2.....	90
Figura 59:	Tempo computacional para cada um dos <i>makespan</i> na fase 2 .....	91

Figura 60:	Gráfico de Gantt com o programa de produção do exemplo 2 na fase 3 do modelo (68)-(75) .....	92
Figura 61:	Variação do número de operadores no gabarito e na bancada para a fase 3 do modelo (68)-(75) .....	93
Figura 62:	Variação do número de operadores no exemplo 2 na fase 3 do modelo (68)-(75).....	94
Figura 63:	Tamanho das equipes de montagem em função do <i>makespan</i> para a fase 3 do modelo (59)-(67) .....	95
Figura 64:	Tamanho ótimo das equipes de montagem em função do <i>makespan</i> para a fase 3 do modelo (68)-(75) .....	95
Figura 65:	Tempo computacional para cada um dos <i>makespan</i> na fase 3 do modelo (59)-(67).....	96
Figura 66:	Tempo computacional para cada um dos <i>makespan</i> na fase 3 do modelo (68)-(75).....	96
Figura 67:	Gráfico de Gantt com o programa de produção do exemplo 2 na fase 4 do modelo (52)-(58) .....	98
Figura 68:	Variação do número de operadores no gabarito e na bancada para a fase 4 do modelo (52)-(58) .....	99
Figura 69:	Variação do número de operadores no exemplo 2 na fase 4 do modelo (52)-(58).....	100
Figura 70:	Tamanho ótimo das equipes de montagem em função do <i>makespan</i> do modelo (52)-(58) .....	100
Figura 71:	Tempo computacional para cada um dos <i>makespan</i> na fase 4 do modelo (52)-(58).....	101

# SUMÁRIO

1	Introdução .....	1
1.1	Objetivo .....	2
1.2	Justificativas .....	3
1.3	Método de pesquisa .....	4
1.4	Organização do trabalho .....	5
2	Indústria aeronáutica.....	6
2.1	Produção de Aviões .....	9
2.2	Fabricação e Montagem.....	12
2.3	Tipos de gabaritos de montagem .....	17
3	Revisão da Literatura.....	20
3.1	Planejamento e controle da produção .....	20
3.2	Classificação dos problemas de programação de produção .....	22
3.3	Programação de <i>job shop</i> .....	24
3.4	Programação de projetos .....	28
3.5	Restrições de adjacência.....	35
3.6	Curvas de Aprendizagem.....	48
4	Descrição do Problema e Modelagem Matemática .....	52
4.1	Apresentação do problema .....	52
4.1.1	Montagem do subconjunto .....	52
4.1.2	Estratégias de produção por avião e por peça.....	58
4.2	Modelo matemático da fase 1 .....	62
4.3	Modelo matemático da fase 2.....	64
4.4	Modelo matemático da fase 3.....	67
4.5	Modelo matemático da fase 4.....	70
5	Resultados Computacionais.....	72
5.1	Resultados computacionais da montagem do subconjunto .....	72
5.1.1	Exemplo computacional da fase 1 .....	72
5.1.2	Exemplo computacional da fase 2 .....	86
5.1.3	Exemplo computacional da fase 3 .....	91
5.1.4	Exemplo computacional da fase 4 .....	97
5.2	Discussões finais .....	102
6	Conclusões e perspectivas .....	104
7	Referências .....	106
8	APÊNDICE A - Tabelas de durações das operações de montagem.....	117



## 1 Introdução

A indústria aeronáutica é conhecida por ser um setor que utiliza alta tecnologia, tanto na pesquisa e desenvolvimento, quanto na produção e operação dos seus produtos. Entretanto, diferentemente de indústrias como a automobilística, os estudos relacionados à produção são predominantemente direcionados a aspectos técnicos como tolerâncias, desenvolvimento de novos materiais e processos de fabricação. Os métodos de produção de aviões do ponto de vista da gerência da produção ainda são relativamente pouco estudados. Isso se deve a diversos fatores, como a grande preocupação no desenvolvimento das características do produto, alta complexidade das operações, baixo volume unitário e consequentemente baixa repetição e produção fundamentalmente manual.

O setor aeroespacial é um dos setores industriais mais importantes do Brasil, sendo um dos maiores exportadores (BRASIL, 2009a), com receita de cerca de US\$ 5 bilhões, com o Brasil ocupando a quinta posição entre os maiores exportadores mundiais, atrás de Estados Unidos, França, Alemanha e Canadá (FERREIRA et al., 2009). Insead (2010) aponta o setor aeronáutico como um dos três mais inovadores na economia brasileira. A indústria aeronáutica está entre os setores em que o governo brasileiro está investindo para aumentar a competitividade, utilizando a Política de Desenvolvimento Produtivo do Ministério do Desenvolvimento e Comércio Exterior. Entre as metas específicas para o setor aeroespacial está a de atingir a produtividade de R\$ 200 mil faturamento/empregado/ano, promover a modernização industrial e aumentar a participação do mercado de jatos executivos, entre outras, o que exige investimentos, cooperação e inovação em pesquisas por parte das empresas e universidades para cumprir esses objetivos. Segundo a Associação das Indústrias Aeroespaciais do Brasil (AIAB), em 2008 o setor empregou cerca de 30 mil pessoas (AIAB, 2010).

Nesse contexto, muitas técnicas de gerência da produção que estão em uso em outras indústrias começam a chamar a atenção da indústria aeronáutica pelo potencial de aumento de eficiência que essas técnicas possuem. A pesquisa operacional é uma das áreas que têm muito a contribuir para a indústria aeronáutica, sendo que uma das técnicas de pesquisa operacional mais conhecida e utilizada nessa indústria é a simulação. Por exemplo, Scott (1994), Dale (2001) e Chikong et al. (2006) utilizaram simulação para estudar distribuição de equipes de montagem, rebitagem automática e mudança de *layout*, respectivamente. Heike et al. (2001), Abuabara (2006) e Abuabara e Morabito (2008; 2009)

estudaram respectivamente a distribuição de equipes de montagem e a otimização de cortes de tubos estruturais utilizando modelos lineares inteiros. Filho e Yoshisaki (2004) e Magalhães (2008) estudaram a cadeia logística da indústria aeronáutica utilizando dinâmica de sistemas e simulação, respectivamente.

A produção de aviões é caracterizada por ser predominantemente manual, especializada e exige um elevado nível de precisão, confiabilidade e qualidade, pois o produto, o avião, não admite defeitos de produção durante a sua operação. Alguns estágios da produção, como a fabricação e algumas montagens, apresentam alguma automação, mas a produção é predominantemente manual. Nos estágios de montagem manual são utilizados equipamentos chamados gabaritos, que podem ter diversos tamanhos e formatos, com a finalidade de garantir que as condições de projeto sejam respeitadas durante a produção.

Nesta indústria, as montagens, como ocorrem em baixa cadência, depois de um número de unidades produzidas, possuem duração mais previsível e podem então, ser estudadas com modelos matemáticos determinísticos de programação de produção. Para abordar os problemas de programação da produção de montagens, é necessário adaptar os modelos da literatura para representar as características específicas presentes nas operações de produção das indústrias aeronáuticas. Como a programação de produção nesta indústria em geral é feita manualmente, existe uma boa expectativa de melhoria em relação ao praticado.

## 1.1 Objetivo

O objetivo desse trabalho é estudar, utilizando modelagem matemática de otimização, as montagens que ocorrem em gabaritos com capacidade de processar múltiplas tarefas ao mesmo tempo, porém, com algumas restrições. Esses são chamados na indústria aeronáutica de gabaritos com restrições de adjacência. O problema é programar as tarefas a serem executadas nos gabaritos, sujeitas às restrições de recursos, como mão de obra e postos de trabalho dos gabaritos, visando minimizar o tempo total de entrega das tarefas, *makespan* ou a quantidade total de recursos utilizados.

Portanto, o trabalho tem como objetivo desenvolver modelos matemáticos para auxiliar na tomada de decisão em nível operacional, definindo a programação de atividades e auxiliando na definição das quantidades de recursos a serem utilizadas em dado período. Por isso, também serve como ferramenta auxiliar para estudo de cenários futuros, avaliando os impactos das variações nas cadências de produção na indústria aeronáutica.

Outro objetivo do trabalho é aplicar os modelos matemáticos em situações práticas e então, avaliar a qualidade da solução e o comportamento de fatores como o nivelamento da mão de obra, o tempo computacional para alcançar a solução e as suas variações para instâncias diferentes do problema.

## 1.2 Justificativas

O estudo da programação de produção é uma alternativa importante dentro das preocupações da gerência da produção. É uma abordagem pouco utilizada na prática da indústria aeronáutica, apesar de já ser conhecida e aplicada em diversos setores. Não foram encontradas na literatura pesquisas específicas tratando de aplicações de modelagem matemática para programação da produção de montagens em gabaritos presentes na indústria aeronáutica, apesar de ser um problema importante dentro do contexto de planejamento e controle de produção de aeronaves, o que indica oportunidades para estudos e aplicações de programação da produção.

A indústria aeronáutica possui características de produção que são relativamente específicas, abrindo oportunidades para o desenvolvimento de novas abordagens de programação da produção. Apesar das especificidades da produção de aviões, essas novas abordagens poderiam ser aplicáveis a outras situações fora do contexto da indústria aeronáutica, caso os problemas, do ponto de vista de programação, sejam parecidos. A utilização de modelos matemáticos na programação da produção em indústria aeronáutica, principalmente em setores que a programação da produção é feita de forma manual, apresenta grandes chances de gerar melhorias significativas na utilização dos recursos.

Em geral, como as montagens são manuais, estas dependem fundamentalmente de pessoas, que nesse caso, são qualificadas e custam caro. A falta de uma programação formal gera dificuldades em administrar o andamento das montagens, identificar retrabalhos e outros problemas de qualidade, planejar horas extras, cumprir prazos de entrega, planejar a quantidade necessária de recursos para atender variações de cadência no futuro, subaproveitamento dos recursos existentes, dificuldade em se adaptar as variações na produção e uma série de outros problemas. Assim, acredita-se que o estudo da programação da montagem de alguns gabaritos é uma pesquisa relevante, do ponto de vista de competitividade, como uma forma de otimizar, sob diversos aspectos, a produção de aviões.

### 1.3 Método de pesquisa

Do ponto de vista de método de pesquisa, as abordagens podem ser classificadas como qualitativa ou quantitativa. Bryman (1989) afirma que a diferença entre as duas abordagens é a perspectiva que será considerada para definir as variáveis de estudo. Segundo Bryman (1989), na pesquisa qualitativa o pesquisador enfatiza a perspectiva e interpretação dos indivíduos envolvidos na pesquisa. O pesquisador deve restringir as pessoas que participam da pesquisa o mínimo possível, de forma a compreender o ponto de vista de todos os envolvidos. Por isso, as pesquisas qualitativas utilizam métodos como entrevistas ou pesquisa participante.

Diferentemente da abordagem qualitativa, a abordagem quantitativa enfatiza pouco a perspectiva das pessoas envolvidas na problemática. O pesquisador formula hipóteses baseado em leis gerais e, assim, escolhe as variáveis que serão estudadas, ou, muitas vezes, as variáveis já estão definidas na literatura (BRYMAN, 1989). Os métodos típicos da abordagem quantitativa são a modelagem e simulação ou a experimentação, entre outros.

Este trabalho utiliza uma abordagem quantitativa por basear a escolha das variáveis na teoria de programação de produção, sendo a participação das pessoas envolvidas no contexto problema, bastante limitada. Além disso, as variáveis do problema são em geral de controle relativamente fácil e dependem fundamentalmente da decisão do responsável pela produção. É responsabilidade do pesquisador entender e decidir quais as preocupações das pessoas envolvidas no problema que devem ser consideradas ou não na pesquisa. O método de pesquisa utilizado neste trabalho é o de modelagem e simulação, segundo as definições de Bertrand e Fransoo (2002) e Morabito e Pureza (2009). A modelagem e simulação utilizam linguagem matemática e computacional para abstrair situações práticas ou idealizadas e assim, poder manipulá-las a partir de técnicas analíticas e experimentais, gerando sugestões de ações ou novos conhecimentos de relações entre as variáveis de pesquisa (MORABITO, PUREZA, 2009).

Essa pesquisa utiliza a abordagem de modelagem por usar modelos matemáticos de otimização, dentro do contexto de pesquisa operacional, para estudar os problemas de programação de gabaritos com restrições de adjacência. Bertrand e Fransoo (2002) e Morabito e Pureza (2009) dividem os problemas de modelagem em axiomáticos ou empíricos, e descritivos ou normativos. Os problemas axiomáticos são problemas idealizados pelo pesquisador. Os modelos empíricos são problemas práticos que o pesquisador se propõe a resolver. Os problemas descritivos se preocupam em descrever o funcionamento do

problema e as relações das variáveis deste, aumentando o conhecimento do problema. Os problemas normativos se preocupam em prescrever soluções e procedimentos para um problema.

Essa pesquisa está em um contexto prático visando melhorar os procedimentos de programação da produção e, por isso, a pesquisa é empírico-normativa.

#### 1.4 Organização do trabalho

Este trabalho está dividido em oito capítulos. O capítulo 1 é introdutório. No capítulo 2 é apresentada a indústria aeronáutica e descrito brevemente o processo de produção de aviões, as partes que compõem os aviões e como é o processo de montagem, com a utilização dos gabaritos de montagem.

No capítulo 3 é revisada a literatura de programação de produção, apresentando os conceitos teóricos básicos e, em seguida, as medidas de desempenho e os modelos matemáticos utilizados nos problemas de programação, como o clássico modelo de *job shop* e os modelos para programação de projetos com restrições de recursos. Nesse capítulo também são apresentados os conceitos de curva de aprendizagem.

No capítulo 4 são apresentados os modelos matemáticos propostos para o problema estudado. São vários modelos, sendo que cada um foi apresentado em uma seção específica.

No capítulo 5 são apresentados os resultados computacionais dos modelos do capítulo 4, apresentando as soluções encontradas para cada um dos modelos apresentados no capítulo 4.

No capítulo 6 são apresentados os resultados e as conclusões que podem ser feitas a partir dos resultados apresentados no capítulo 5. No capítulo 6 também são apresentadas perspectivas de pesquisa para o problema.

## 2 Indústria aeronáutica

Apesar de não reconhecido em alguns países, o primeiro avião foi inventado e testado com sucesso em público por um brasileiro chamado Santos Dumont, em 23 de outubro de 1906 (DUMONT, 1918). Seu invento foi o primeiro a decolar, voar e pousar por meios próprios. Santos Dumont era um inventor nato e possui uma série de outras invenções importantes em uso até hoje, mas nunca se preocupou em comercializar ou proteger as suas invenções por patentes. Embora tivesse recebido fortes críticas em sua época, ele acreditava que o avião iria inaugurar uma grande era e indústria no futuro, quando todos teriam seus próprios aviões.

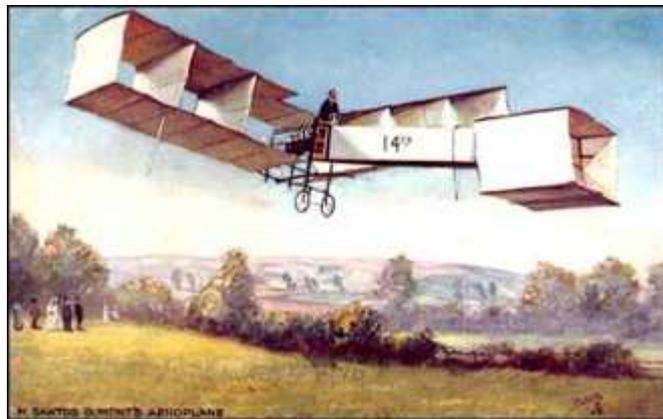


Figura 1: Vôo do 14 Bis, Campo Bagatelle, Paris (NASA, 2006).

Nem todas as visões de Santos Dumont para o futuro se concretizaram ainda, não se sabe se por elas estarem erradas ou se por serem demasiado avançadas para que apenas pouco mais de um século fosse suficiente para realizá-las. De fato, o mercado de aviação no mundo, hoje, é gigantesco e foi responsável por cerca de US\$ 101 bilhões em 2006, com a projeção de movimentar US\$ 112,5 em 2015 (TEAL GROUP, 2006), incluindo aviação civil e militar. Isso significa cerca de US\$ 550 bilhões nos próximos 5 anos. É prevista, entre substituição de aeronaves e aumento da frota, a demanda de 6.750 aeronaves de 30 a 120 assentos, categoria em que a Embraer possui diversos produtos, e 14.650 aeronaves de 121 até 210 lugares entre 2009 e 2028 (EMBRAER, 2010). Portanto, é um mercado importante e com grande potencial para a indústria aeronáutica brasileira.

Além disso, a indústria aeroespacial utiliza de forma intensiva pesquisa e desenvolvimento, emprega operários qualificados e grande quantidade de engenheiros e cientistas. Por essas razões, é um segmento que produz tecnologia de ponta que é transferida para outros setores da economia, gerando muitos outros benefícios indiretos (SEITZ; STEELE, 1985). Também é um setor muito próximo a questões de defesa e aeroespaciais,

sendo por todos esses fatores, um setor estratégico para todos os países, principalmente para as potências mundiais.

O Brasil participa desse mercado, com tecnologia e uma cadeia de produção pequena em relação às duas maiores empresas do setor (a americana Boeing e a européia EADS), mas crescente e bastante significativa tanto economicamente quanto simbolicamente para o país. A principal empresa brasileira do setor é a Embraer e seus produtos são predominantemente exportados. Conforme apresentado na figura 2, as exportações da Embraer saltaram de US\$ 1,5 bilhão em 1999 para quase US\$ 6 bilhões em 2008, e provavelmente continuará crescendo, pois a empresa está expandindo seu portfólio de produtos para o mercado de aviação executiva (EMBRAER, 2008; 2009). Insead (2010) aponta a indústria aeronáutica como um dos três setores mais inovadores e com tecnologia de classe mundial do Brasil (os outros são a exploração de petróleo em águas profundas e agricultura tropical).

Além disso, a empresa enfrentará novos concorrentes da Rússia, Japão e China, no mercado de jatos regionais até 120 assentos (OTTOBONI, 2007; SUKHOI, 2007; OLMOS, 2008; MECHAM, ANSELMO, 2008; BOWKER, 2009). No ano de 2006 os Estados Unidos possuíam 54,3% do mercado, enquanto a Europa possuía 35,1% e o restante do mundo 10,6% (TEAL GROUP, 2006).

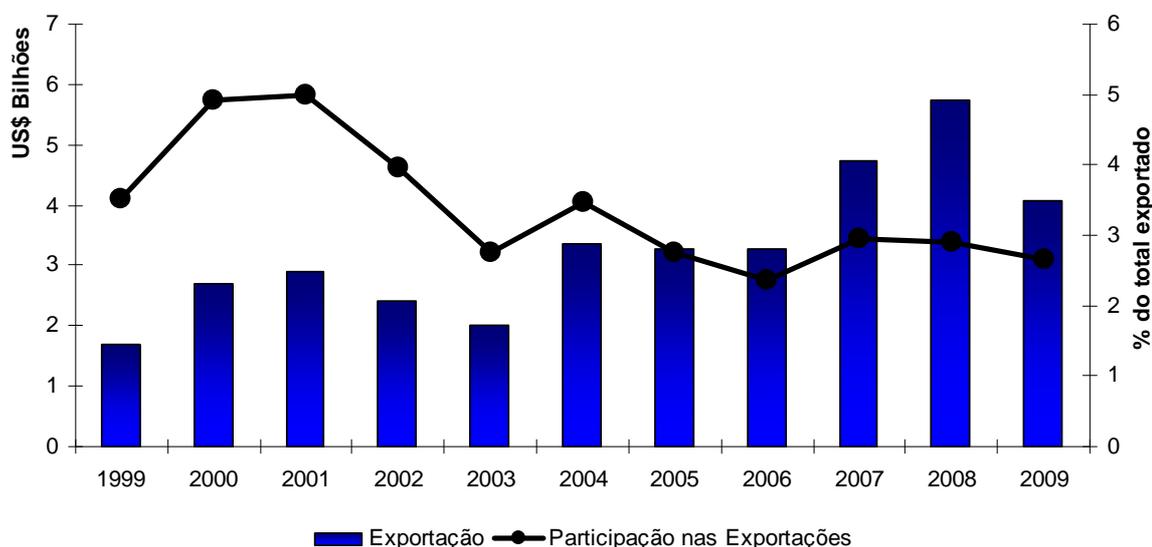


Figura 2: Exportação da Embraer em US\$ bilhões nos últimos 10 anos (BRASIL, 2009a).

No ambiente de maior importância e maior competitividade, as pesquisas da indústria aeronáutica não estão mais apenas focadas em aspectos de desenvolvimento de produtos e tecnologias aeronáuticas, mas também, em aspectos de produção. Uma das maiores

iniciativas no mundo é o *Lean Aerospace Initiative* (L.A.I.), uma rede de pesquisadores de universidades, profissionais e funcionários do governo que pesquisam técnicas de manufatura enxuta, originadas na indústria automobilística, aplicadas ao projeto e produção de aviões. É uma rede liderada pelo *Massachusetts Institute of Technology* (M.I.T.) e envolve as principais empresas e órgãos do governo ligados ao setor aeroespacial dos Estados Unidos. O *Lean Aerospace Initiative* é parte de um grupo de pesquisa mais amplo também liderado pelo M.I.T., conhecido como *Lean Advancement Initiative* (L.A.I.), ambos com a mesma sigla, que possui as mesmas atribuições do *Lean Aerospace Initiative*, mas para toda a indústria dos Estados Unidos.

No Brasil também é feita pesquisa na indústria aeronáutica, embora de modo consideravelmente menos estruturado do que nos Estados Unidos. As pesquisas estudam aspectos históricos da formação da indústria aeronáutica brasileira (BOTELHO, 1999; BERTAZZO, 2003; FORJAZ, 2004), aspectos ergonômicos (SECCHIN, 2007; SOUZA, 2008), gestão do conhecimento (SANTOS; NETO, 2008) ou gestão do desenvolvimento e projeto de produtos (FREIXO, 2004; BARBOSA, 2007).

Em uma pesquisa realizada na base *web of knowledge*, poucos trabalhos foram encontrados utilizando técnicas de otimização matemática aplicadas a problemas de manufatura da indústria aeronáutica. Scott (1994) estuda a alocação de equipes de montadores na montagem final de jatos comerciais utilizando simulação. A programação de rebitoras automáticas foi estudada e aplicada em Dale (2001), também utilizando simulação. Chikong et al. (2006) utiliza simulação para estudar a mudança para o *layout* celular de uma linha de produção de revestimentos de estruturas aeronáuticas. Heike et al. (2001) estuda a alocação de equipes de montagem em linhas de montagem de aviões, considerando equipes com diferentes capacidades e diversas estratégias de atendimento utilizando modelos matemáticos lineares inteiros e não-lineares. Abuabara (2006), Abuabara e Morabito (2008; 2009) estudaram a otimização de cortes de tubos estruturais em aeronaves agrícolas utilizando modelos lineares inteiros. A manufatura na indústria aeronáutica, com base na experiência do próprio autor, é um terreno fértil para aplicação de diversos problemas conhecidos da pesquisa operacional, como balanceamento de linhas de produção, programação de mão de obra para montagem de subconjuntos, programação da produção, corte e empacotamento, teoria de filas, entre vários outros.

## 2.1 Produção de Aviões

Aviões podem ser divididos em partes que possuem funções e características específicas. Genericamente, a estrutura dos aviões pode ser dividida em três partes: fuselagem, asas, e empenagem, formada pelos estabilizadores vertical e horizontal (HOWE, 2004). Além disso, o avião possui equipagens como trem de pouso e motores que podem ser projetadas em várias posições nos aviões, conforme as características desejadas. A figura 3 ilustra as principais partes que compõem o avião.

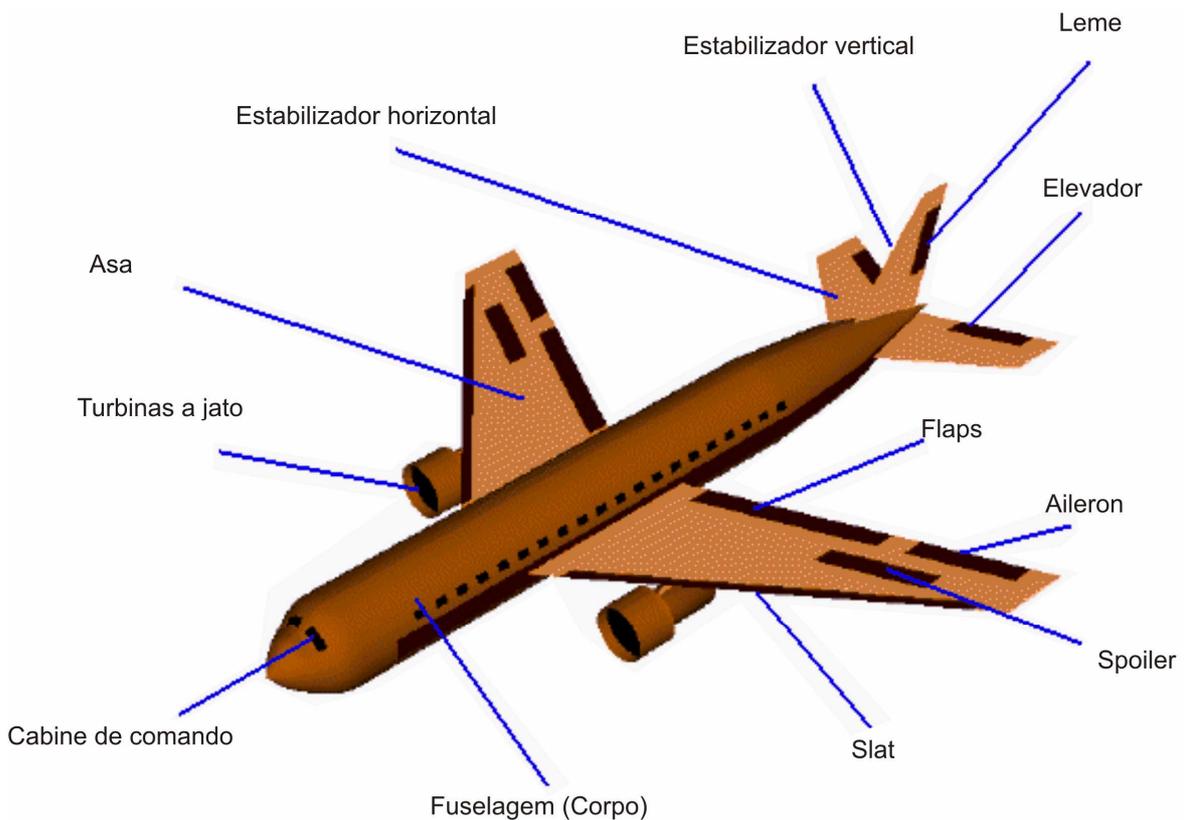


Figura 3: Principais partes que compõem o avião (NASA, 2008).

A fuselagem possui formato genérico de um cilindro, formado por diversos segmentos. Seu formato não é exatamente cilíndrico por que a seção transversal da fuselagem não é exatamente um círculo. Ela é influenciada por características de configuração interna (como assentos de passageiros), tamanho do avião e características aerodinâmicas. A figura 4 apresenta um exemplo de fuselagem e o respectivo formato da seção transversal.



Figura 4: Fuselagem seção transversal (BOMBARDIER AEROSPACE, 2009).

Os segmentos da fuselagem são formados por painéis metálicos geralmente de alumínio, que são moldados internamente por uma estrutura em formato de teia, que tem a função estrutural da fuselagem. É também nos painéis da fuselagem que são feitos os recortes que formam as janelas, portas e pontos de acesso para manutenção. A figura 5 apresenta como é formada a estrutura dos aviões.

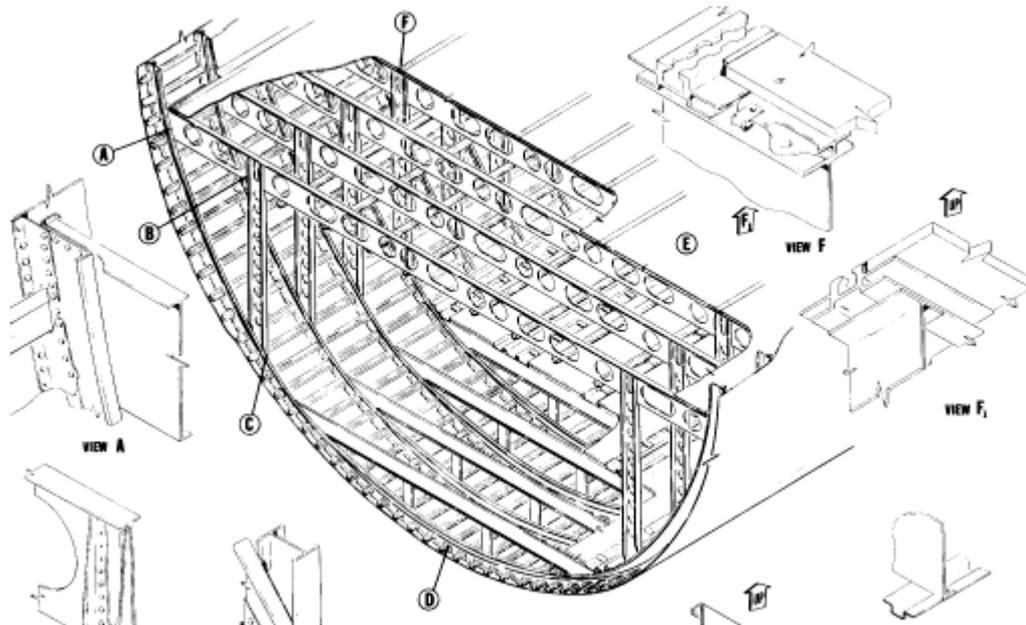


Figura 5: Painéis e estruturas de cavernas e vigas omega (STANFORD, 2009).

As asas estão montadas lateralmente na fuselagem e possuem como principal função gerar sustentação para que toda a aeronave possa voar. Além de serem responsáveis pelo vôo, as asas podem ter outras funções, como carregamento de carga em aviões de combate, alojar os trens de pouso principais, carregar as turbinas, tanques de combustível, armamentos e auxiliar no controle do vôo por meio dos *ailers* ou auxílio no pouso com os

*flaps* e freios aerodinâmicos. As asas são conectadas à fuselagem pelo *wing stub*, que é um componente estrutural que faz a junção das asas com a fuselagem. A figura 6 ilustra um exemplo de asa e alguns componentes estruturais.

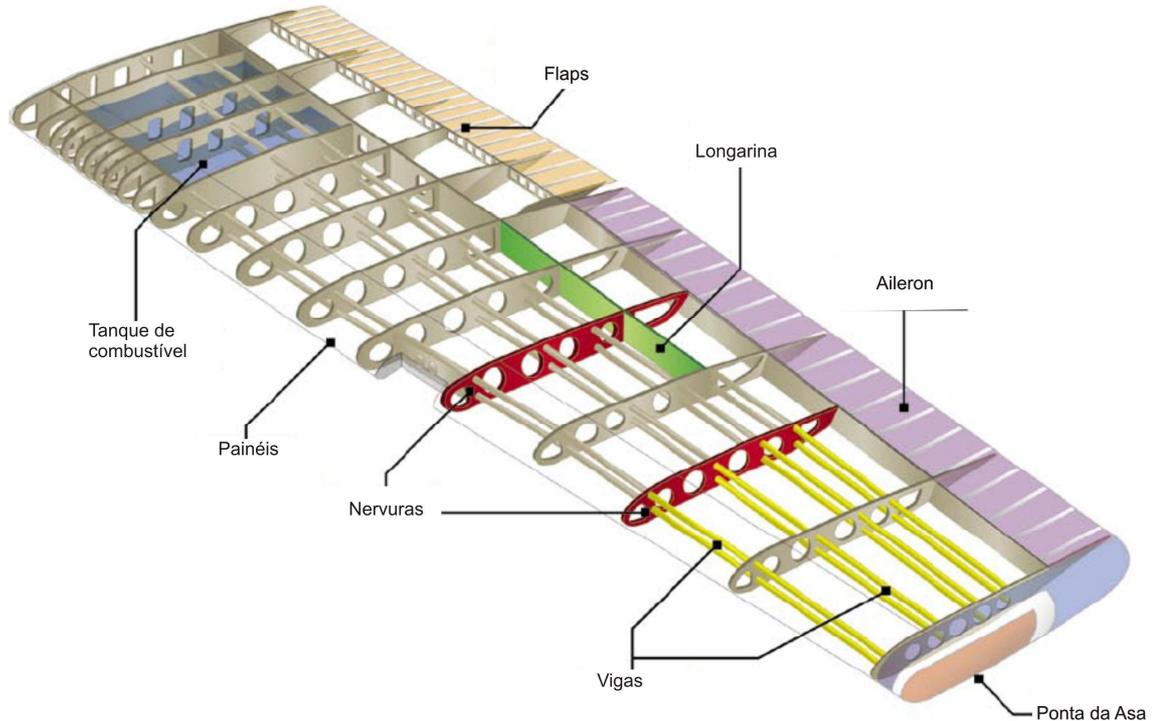


Figura 6: Exemplos de asas e componentes (U.S.A., 2008).

Por último, a empenagem, que fica na parte traseira do avião, é formada pelos estabilizadores vertical e horizontal, leme e profundores. Estes atuam basicamente no controle do voo, controlando a inclinação e direção do avião. Suas configurações também variam conforme as características desejadas do avião.

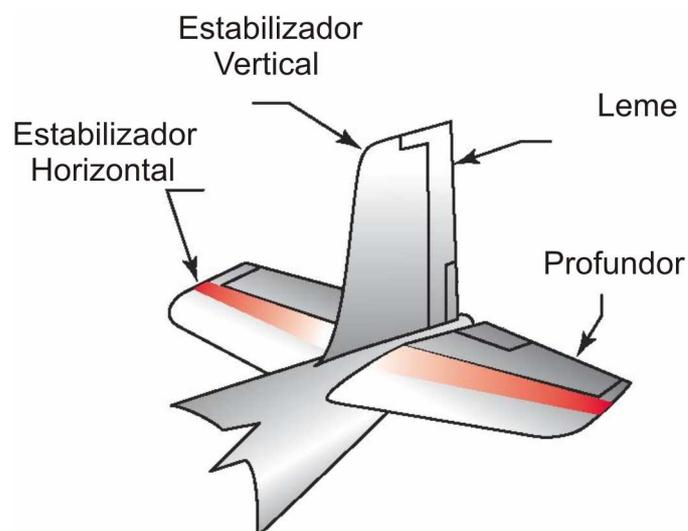


Figura 7: Empenagem (U.S.A., 2003)

## 2.2 Fabricação e Montagem

No início do século XX, a nascente indústria aeronáutica fabricava aviões basicamente de madeira. Embora em desuso, a madeira apresenta propriedades mecânicas comparáveis às de ligas de alumínio moderna (NIU, 1988). Os processos de fabricação com madeira são baratos, mas a madeira apresenta desvantagens como resistência ao calor baixa e sensibilidade às variações de umidade, além de estar sujeita a ataques de cupins e fungos.

Durante o século XX, foram desenvolvidas uma série de ligas metálicas que compõem a estrutura dos aviões, com características específicas para atender aos requisitos dos aviões em desenvolvimento. As três ligas metálicas que se destacam para a utilização em estruturas aeronáuticas são as ligas de alumínio, de aço e de titânio (MEGSON, 1999). As ligas de alumínio podem ser compostas de diversos elementos e quantidades diferentes, mas possuem de forma geral, boa adaptabilidade aos requisitos de desempenho estrutural (como resistência à propagação de fissuras, ductibilidade, alta dureza, resistência à corrosão) e de fabricação aliado a um baixo peso. Por isso, as ligas de alumínio são utilizadas largamente nas estruturas aeronáuticas.

As ligas de aço possuem um peso específico grande e por isso, a sua utilização nos aviões é restrita a partes específicas. São utilizadas quando é necessário suportar grandes tensões. Por essa característica, são de difícil fabricação e precisam de processos especiais de conformação e usinagem (MEGSON, 1999). As ligas de titânio apresentam propriedades altamente específicas, mas, de modo geral, possuem propriedades que não variam, mesmo para altas temperaturas, porém, apresentam elevados pesos específicos e custos de produção.

Outros materiais como vidros, plásticos e materiais compostos (fibra de carbono, aramida, etc.) também são utilizados nas estruturas aeronáuticas (MEGSON, 1999). Como os principais materiais dos aviões são ligas metálicas, os principais processos de fabricação estão relacionados à fabricação na indústria metal mecânica, como usinagem, tratamentos térmicos e de superfícies, corte, conformação, solda, fundição, entre outros (GROOVER, 2007).

Tipicamente, a fabricação na indústria aeronáutica utiliza dois principais processos de fabricação, que são a usinagem e a estampagem. A usinagem é o processo de transformar blocos, ou seja, metais com mais de meia polegada de espessura, em peças com o formato desejado. Diversas tecnologias de usinagem podem ser utilizadas, desde os tradicionais tornos até usinagem química, dependendo das características desejadas na peça

em produção. Os blocos usinados podem variar de pequenas peças de alguns centímetros a grandes peças de mais de 1 metro.

A estampagem é a transformação de chapas metálicas em peças, utilizando basicamente alguma tecnologia de conformação, como prensas ou calandras. O processo depende também das características específicas da peça a produzir e pode incluir outros processos sofisticados, como tratamentos térmicos ou tratamentos de superfície. As peças estampadas podem ter desde alguns poucos centímetros, como peças de subconjuntos pequenos, até vários metros, como os grandes revestimentos que compõem a parte externa dos aviões.

Com as peças fabricadas, pode-se então, fazer as montagens. A montagem mecânica é feita basicamente utilizando diversos tipos de parafusos e rebites com diferentes propriedades e algumas pequenas variações nos processos de montagem que fixam as peças componentes e formam os subconjuntos e conjuntos (GROOVER, 2007). As montagens na indústria aeronáutica variam em tamanho, duração e complexidade. As montagens menores e de menor complexidade são as de subconjuntos. As montagens de maior complexidade são as montagens estruturais, pois envolvem a montagem de diversos subconjuntos.

Os subconjuntos são quaisquer duas ou mais peças componentes que foram montadas. Os subconjuntos de montagem pouco complexa são montados manualmente, sobre bancadas, utilizando ferramentas manuais e algumas vezes processos especiais, como hidrogênio líquido para a instalação de buchas. Secchin (2007) e Souza (2008) descrevem com detalhes algumas operações de montagem que são utilizadas em montagens de subconjuntos e montagens estruturais.

Como qualquer montagem, as que ocorrem na indústria aeronáutica devem estar preocupadas em garantir que os subconjuntos montados estejam dentro das tolerâncias especificadas em projeto e assim, possuírem a qualidade e o desempenho previstos em projeto. Como os processos de montagem são manuais, em geral, é utilizada uma tecnologia de montagem em gabaritos que assim, garantem as condições de projeto. A figura 8 apresenta um exemplo de montagem de um subconjunto com os respectivos gabaritos de montagem ao fundo. Repare que o operador está realizando uma operação de montagem na bancada.



Figura 8: Montador e o respectivo gabarito de montagem de um subconjunto (AERO, 2009).

Os gabaritos são estruturas geralmente metálicas, fixas, e projetadas para não variar as dimensões conforme a temperatura, forças mecânicas ou peças em montagens presentes no ambiente de produção. Devem ser alinhados, para assegurar que consigam garantir as condições de projeto. Os gabaritos são utilizados na indústria metal mecânica em geral, porém, estão presentes na indústria automobilística e principalmente na indústria aeronáutica mundial.

Os gabaritos são utilizados para montagens manuais em decorrência da inviabilidade técnica e/ou econômica de automação do processo de montagem. No caso da indústria aeronáutica, embora existam diversos equipamentos CNC (equipamentos com controle numérico computadorizado) que fazem montagens em funcionamento ou desenvolvimento pelo mundo (HARTMANN et al., 2000; BUCKINGHAM et al., 2006; FAYAWEERA; WEBB, 2007; DEVIEGLI; FEIKERT, 2008), eles não substituem as operações manuais totalmente.

O principal motivo que impede a automação da montagem das estruturas aeronáuticas é o acesso a todas as partes que precisam ser montadas e por isso, são utilizadas pessoas, pois são mais flexíveis. Existem esforços consideráveis na utilização de robôs com grande capacidade de articulação, mas a sua utilização ainda é específica a alguns subconjuntos. Crowson (2006) afirma também que a baixa automação na indústria aeronáutica é consequência da produção em baixo volume e, quando existente, se restringe às operações que exigem alto grau de precisão.

Riley (1996) comenta que a automação é adequada para produção altamente repetitiva, sofrendo forte competição internacional ou intensiva em mão de obra. Os dois últimos fatores ocorrem na indústria aeronáutica. A automação de montagens na indústria aeronáutica é um ramo promissor de avanço da tecnologia de produção. Cabe ressaltar que o

Brasil tem participado desse esforço através, por exemplo, do LAME (Laboratório de Automação de Montagem Estrutural), um laboratório de pesquisa pertencente ao Centro de Competência em Manufatura, ligado ao ITA (BRASIL, 2009b).

As conseqüências das dificuldades de automação na indústria aeronáutica são que as montagens são caracterizadas por utilizar amplamente gabaritos de montagem, com tamanhos que variam de 30 cm a vários metros, montagens geralmente manuais e que utilizam de uma até dezenas de pessoas, e acontecem em postos de trabalho fixos no espaço. Dependendo das características da montagem, cadência de produção e viabilidade técnica, pode-se utilizar vários gabaritos de montagem seqüenciais ou vários iguais, combinados com a utilização de montagens em bancadas e equipamentos CNC ou robôs. A figura 9 apresenta um painel da fuselagem sendo montado em um equipamento de rebiteagem CNC. A utilização desses equipamentos não elimina a necessidade de operações manuais de montagem.



Figura 9: Máquina CNC que faz a rebiteagem de painéis (BRÖTJE, 2009).

Com os subconjuntos montados, pode-se então fazer as montagens estruturais, que utilizam justamente esses subconjuntos. Geralmente, os gabaritos de montagem estrutural são maiores, podendo ter diversos metros de comprimento e vários níveis de altura, inclusive com várias plataformas de apoio. A figura 10 ilustra a montagem estrutural do Boeing 787 *Dreamline*. A montagem estrutural pode incluir também a junção das partes até então

montadas, como as asas e as partes da fuselagem. Nessa fase, como as partes do avião ganham dimensões maiores, geralmente se utiliza equipamentos de movimentação como pontes rolantes, mesmo para aviões de menor porte como aviões de combate e pequenos jatos executivos. Algumas empresas pintam a fuselagem e as asas separadamente. Outras fazem a montagem e junção das partes e das asas e então, pintam o avião.



Figura 10: Montagem estrutural de um Boeing 787 (BOEING, 2009).

Na fase de montagem estrutural também pode ser feita uma pré-equipagem, que nada mais é do que montar no avião alguns equipamentos básicos, como sistemas hidráulicos, cablagens elétricas e cabos de comando de vôo. Na fase de equipagem e montagem final então, monta-se os motores, o interior do avião com o acabamento adequado para o caso de aviões comerciais ou executivos e todos os equipamentos mais sensíveis como os aviônicos e os trens de pouso. Nesta etapa, são realizados diversos testes em solo, visando inspecionar as condições de funcionamento dos equipamentos instalados. A produção então realiza a última etapa, que são os chamados vôos de produção. São vôos realizados pelo fabricante com a finalidade de verificar características chave no avião e se certificar que ele está pronto para ser operado pelo cliente.

A figura 11 apresenta as etapas de produção descritas anteriormente. Os grandes fabricantes de aviões em geral, não trabalham sozinhos. Os estudos dos mercados potenciais e as estratégias de lançamento de produtos, ocupação de nichos de mercado e

outras características mercadológicas são definidas pelos grandes fabricantes. Porém, a produção e algumas partes do projeto dos aviões são divididas com empresas parceiras do projeto, no modelo conhecido como divisão de riscos.

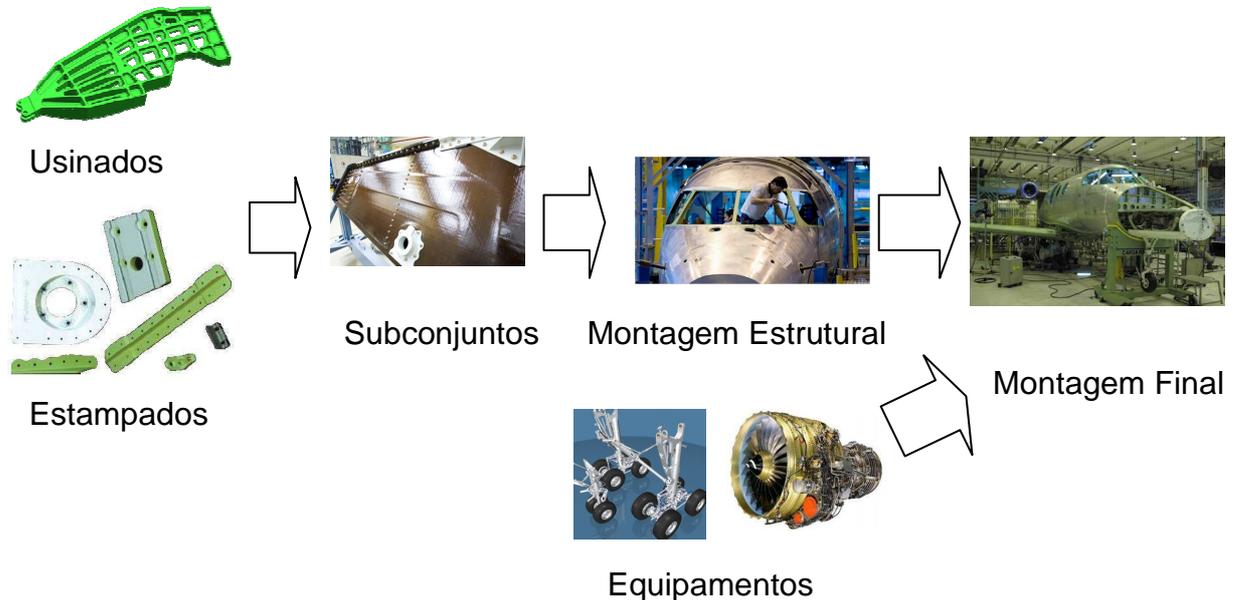


Figura 11: Etapas de produção dos aviões

Assim, em geral, as grandes empresas aeronáuticas do mundo dominam e realizam predominantemente as montagens finais e grandes montagens estruturais e são responsáveis por integrar todos os outros componentes que são produzidos e projetados por outras empresas. Mas a fabricação, montagens de alguns subconjuntos, algumas montagens estruturais e principalmente o projeto e produção dos equipamentos são, em níveis diferentes para cada empresa, realizados pelas empresas parceiras, dentro do modelo de parceria de risco.

### 2.3 Tipos de gabaritos de montagem

Como já discutido anteriormente, os gabaritos são usados em diversas montagens e podem ter diversas configurações, tamanhos e complexidades. Sua finalidade é garantir que as partes dos aviões em montagem atendam rigorosamente as condições de projeto, além de garantir que as peças componentes sejam intercambiáveis com outros subconjuntos, que são montados em outros gabaritos.

Do ponto de vista de operação da produção, existem três tipos de gabarito. O primeiro tipo são os gabaritos que podem montar somente um tipo de conjunto ou subconjunto de cada vez e em específico. São os gabaritos dedicados a um produto em específico. Possuem assim, baixa flexibilidade e sua programação é relativamente simples. Os

gabaritos dedicados são geralmente os gabaritos com montagens mais complexas, como os gabaritos de montagem das asas ou de grandes subconjuntos específicos, como a cauda ou os estabilizadores (ver figura 3), devido à alta especificidade e complexidade dessas montagens. A figura 12 ilustra um gabarito dedicado para a montagem do estabilizador horizontal. Trata-se de um gabarito relativamente grande e uma montagem complexa.



Figura 12: Gabarito dedicado para montagem de um estabilizador horizontal (PILATUS, 2009).

A programação desse tipo de gabarito deve estar preocupada em garantir que o tempo de ciclo da sua montagem seja compatível com o tempo de ciclo de montagens em etapas anteriores e posteriores, com a finalidade de garantir sincronia. O segundo tipo de gabarito são aqueles que podem montar produtos diferentes, porém, apenas um de cada vez. Alguns conjuntos, tipicamente os conjuntos que formam a fuselagem, são muito parecidos e podem ser montados nos mesmos gabaritos. Para isso, o gabarito tem a capacidade de fazer algumas modificações na sua configuração para se adequar ao produto a ser montado. Essa mudança pode ser feita manualmente e na maior parte das vezes, não leva mais do que alguns minutos. Os gabaritos com essa capacidade são os gabaritos flexíveis. Eles garantem uma flexibilidade importante na produção, economizando espaço na fábrica e investimentos em gabaritos de montagem, pois diminuem a necessidade de outros gabaritos para outros produtos e facilitam a adaptação a variações de cadência.

A figura 13 ilustra um gabarito flexível de montagem de painéis da fuselagem. Esses gabaritos têm a capacidade de montar painéis de diversos tamanhos, porém, apenas um de cada vez.



Figura 13: Gabarito flexível, para montagem de painéis (EVOLUTION MAGAZINE, 2006).

Por último, existem os gabaritos com capacidade de realizar várias montagens ao mesmo tempo. Esses gabaritos são os gabaritos múltiplos. Tipicamente, partes do avião que são adjacentes e que, em algum momento serão unidas por uma montagem para formar um subconjunto maior, podem ser montadas em gabaritos múltiplos. Esses gabaritos exploram justamente a semelhança que partes adjacentes do avião possuem, como as diversas seções da fuselagem.

Esse trabalho estuda especificamente um caso de gabarito múltiplo. Em alguns gabaritos múltiplos os diversos produtos que foram projetados para utilizar o gabarito podem ser montados livremente ao mesmo tempo. No entanto, alguns gabaritos múltiplos conseguem montar apenas configurações específicas ao mesmo tempo, devido a limitações de espaço físico no gabarito. Esses gabaritos são os gabaritos com restrições de adjacência.

A restrição de adjacência surge porque os gabaritos múltiplos montam seus diversos produtos geralmente em postos de trabalho, que são dispostos de forma adjacente. Entretanto, nos gabaritos com restrições de adjacência, enquanto uma montagem está acontecendo em um posto de trabalho, as montagens adjacentes, tanto imediatamente à esquerda quanto imediatamente à direita do posto de trabalho, não podem ser realizadas, pois não há espaço físico para realizar todas as montagens ao mesmo tempo. Porém, outros postos de trabalho que não são adjacentes ao posto de trabalho da montagem que está ocorrendo podem ser utilizados.

### 3 Revisão da Literatura

#### 3.1 Planejamento e controle da produção

O planejamento e controle da produção é a área da empresa preocupada em conciliar as suas atividades produtivas, os fornecedores e os seus respectivos suprimentos e as demandas dos clientes. Tudo isso deve ser feito da melhor forma possível, com a finalidade de garantir a competitividade da empresa. Slack (2002) afirma que o planejamento é o exercício de imaginar situações futuras desejáveis e então, elaborar planos para alcançá-las. Os planos são, portanto, uma intenção de futuro e não necessariamente vão acontecer.

Pinedo (2005) define a programação como “um conjunto de técnicas matemáticas e métodos heurísticos preocupados em alocar recursos limitados para as atividades que devem ser executadas”. Leung (2004) define programação como aquela que está “preocupada com a alocação de recursos escassos a atividades, com o objetivo de otimizar uma ou mais medidas de desempenho”. Morton e Pentico (1993) definem programação de modo um pouco mais abrangente, como “o processo de organizar, escolher e temporizar os usos de recursos para carregá-los com todas as atividades necessárias para produzir as saídas no tempo desejado, enquanto satisfaz um grande número de restrições de tempo e relações entre as atividades e os recursos”.

Pinedo (2008) comenta que é atribuição da programação “lidar com a alocação de recursos às tarefas sobre um dado período de tempo e seu objetivo é otimizar um ou mais objetivos”. Krajewski e Ritzman (1996) definem programação como aquela que “aloca recursos no tempo para completar tarefas específicas”. Eles ainda ressaltam que a programação depende muito das pessoas envolvidas no problema em questão porque são elas que detêm o conhecimento de todo o contexto do problema. Morton e Pentico (1993) também afirmam que, apesar dos progressos que a teoria de programação de produção fez principalmente com o avanço da computação e dos *softwares* que permitiram a automatização de procedimentos, uma grande parcela das atividades práticas de programação de produção ainda depende da capacidade e criatividade das pessoas responsáveis pela programação da produção. Para isso, é necessário conciliar o conhecimento das pessoas envolvidas nas questões de programação da produção com a teoria de programação da produção e transformá-las em linguagem matemática adequada, para que os computadores realizem as tarefas computacionais e os cálculos envolvidos.

Morton e Pentico (1993) afirmam que se isso for feito, as pessoas poderão então, focar seus esforços no conhecimento do contexto da problemática em questão, tarefa

que os modelos matemáticos não conseguem realizar, e assim, analisar questões muito mais sutis e sofisticadas da problemática, que são ignoradas quando a programação é toda feita pelas pessoas, de forma manual. Portanto, a programação da produção não envolve apenas modelos matemáticos e heurísticas, e não apenas o conhecimento das pessoas envolvidas na problemática, mas uma interação entre esses dois aspectos. Com essa ressalva, a definição adotada nesse trabalho é justamente a definição de Morton e Pentico (1993), citada anteriormente, por considerá-la a mais completa.

As definições de Pinedo (2005; 2008), Leung (2004) e Krajewski e Ritzman (1996) enfatizam apenas o caso de alocar recursos às tarefas e, de certa forma, negligenciam o caso inverso, além de também reduzirem a programação da produção a técnicas matemáticas e heurísticas assim, não enfocando a importante interação entre o conhecimento das pessoas e as técnicas matemáticas.

É importante ressaltar que existe uma diferença sutil entre a programação da produção e o sequenciamento da produção. De acordo com a definição adotada, a programação da produção está preocupada com a distribuição das tarefas ou dos recursos ao longo do tempo. O sequenciamento se preocupa apenas com a ordem de execução das tarefas e a sua permutação, sem distribuir as tarefas ao longo do tempo. Assim, pode-se dizer que o sequenciamento é o caso da programação da produção em que não se está preocupado com a distribuição das tarefas ou dos recursos ao longo do tempo.

Leung (2004) faz uma breve revisão da história da evolução da teoria de programação. Segundo ele, a origem da programação da produção está na década de 1950, quando os pioneiros da pesquisa operacional se depararam com os problemas de programação em manufatura. Na década de 1960, alguns cientistas da computação encontraram problemas de programação em sistemas operacionais. Durante esses anos iniciais, como as capacidades computacionais dos computadores eram baixas, os problemas possíveis de serem estudados eram muito limitados, comparando-se com a diversidade de problemas estudados atualmente. Foi nesse período que surgiram os trabalhos clássicos de Jackson (1956) e Johnson (1954). Leung (2004) afirma ainda que foi na década de 1970 que foi provado que muitos problemas de programação eram NP - difíceis.

Da década de 1980 para os dias de hoje, a teoria de programação se expandiu rapidamente, incluindo problemas mais sofisticados e novas abordagens e aproveitando os avanços tanto nas capacidades dos computadores quanto nas dos *softwares*. Potts e Strusevich (2009) apresentam a evolução da teoria de programação da produção nos últimos 50 anos de forma bastante detalhada.

### 3.2 Classificação dos problemas de programação de produção

Os problemas de programação de produção possuem várias classificações. A mais usual é a classificação por quantidade e variedade de máquinas. Os problemas de programação podem ser de única máquina, máquinas paralelas e múltiplas máquinas. Os objetivos a serem otimizados variam e podem ser, por exemplo, minimizar o tempo total para completar as tarefas, o *makespan*, o atraso total, o atraso máximo ou vários outros objetivos, inclusive múltiplos objetivos ao mesmo tempo.

Os problemas de única máquina são aqueles em que é necessário programar as tarefas a serem executadas por uma única máquina. Os problemas de máquinas paralelas são aqueles que as tarefas podem ser executadas em qualquer uma de um conjunto de máquinas idênticas. Os problemas de múltiplas máquinas são aqueles que as tarefas precisam de um conjunto de máquinas diferentes entre si para serem executadas. Nesse caso, o fluxo de operações pelas máquinas pode seguir o padrão de fluxo *flow shop*, *job shop* ou *open shop*. Morton e Pentico (1993) adicionam os *batch shop*. No padrão *flow shop*, todas as tarefas utilizam as mesmas máquinas na mesma ordem. No padrão *job shop*, as tarefas podem utilizar todos os equipamentos e quantas vezes precisarem, seguindo um roteiro de produção previamente definido e conhecido. No *open shop*, como no caso do *job shop*, as tarefas podem utilizar vários equipamentos diversas vezes, no entanto, os roteiros não são previamente definidos e podem ser escolhidos, ou seja, os roteiros são abertos. O *batch shop*, segundo Morton e Pentico (1993), é um caso do *open shop* em que é necessário utilizar grandes lotes de produção para ter ganhos de escala e evitar os custos de duplicar estoque em processo entre a produção final e grandes clientes.

Além desses modelos clássicos, outros problemas que não se encaixam necessariamente nos casos apresentados anteriormente também são estudados. São os problemas de dimensionamento e programação de lotes, que poderiam estar no contexto dos *batch shop*, e os problemas de programação de projeto. Hax e Candea (1984) também consideram os problemas de balanceamento de linha como uma classe de problemas de programação.

Os problemas de dimensionamento de lotes estão preocupados em definir qual o tamanho ótimo de um lote a ser produzido em um sistema de produção, buscando equilibrar o *trade-off* entre o custo para cada lote a ser processado, geralmente um *setup*, custo de estoque em processo e entregas em atraso. Em alguns casos, uma vez definido o tamanho e produto do lote, o problema pode não ser apenas definir quanto e o que produzir, mas também

quando e em que equipamentos, visando otimizar algum objetivo, caracterizando um problema de programação da produção (QUADT, 2004).

Os problemas de programação de projetos são aqueles em que se deve programar grandes conjuntos de tarefas de acontecem juntas apenas uma vez, como grandes obras de infra-estrutura ou projetos de produtos complexos. Recentemente, vários autores estão separando os problemas de programação de máquinas dos problemas de programação de projetos. Rodammer e White (1988) e Potts e Strusevich (2009), em seus respectivos *surveys* sobre programação de produção, separam os problemas de programação de projetos e programação de máquinas em classes distintas. Morton e Pentico (1993), Nahmias (2008) e Pinedo (2005) fazem essa distinção também. A tabela 1 apresenta as principais diferenças entre os modelos de programação de máquinas e programação de projetos.

Tabela 1: Principais diferenças entre os modelos de programação de máquinas e programação de projetos

<b>Programação de máquinas</b>	<b>Programação de projetos</b>
Quantidade de recursos fixos	Quantidades de recursos podem variar
Não consome recursos	Pode consumir recursos
As tarefas utilizam apenas um recurso por vez	As tarefas podem utilizar diversos recursos ao mesmo tempo
Não considera o <i>trade-off</i> de custos entre o tempo e a quantidade de recursos utilizados	Pode considerar o <i>trade-off</i> entre o custo de tempo e de recursos utilizados
Auxiliam decisões tipicamente operacionais	Podem auxiliar decisões táticas e até estratégicas

Os problemas de balanceamento de linhas estão preocupados em distribuir um conjunto de tarefas a serem executadas a um conjunto de recursos, dispostos em linha e com fluxo de produção *flow shop*, de modo a atender um tempo de ciclo máximo (HAX; CANDEA, 1984). O tempo de ciclo da linha é o intervalo de tempo entre o início ou o término de dois produtos consecutivos. A distribuição das tarefas pelas estações de trabalho pode estar sujeita a vários tipos de restrições, como restrições precedência ou tecnologia.

Exceto os problemas de programação de projetos, todos os outros problemas apresentados anteriormente pressupõem que cada tarefa necessitará de apenas um único recurso cada vez que for processada. Blazewicz et al. (1996) apresentam problemas de múltiplos processadores em que as tarefas precisam de mais de um recurso ao mesmo tempo

para ser executadas. Eles dividem essa classe de problemas em dois tipos: os recursos paralelos e os recursos dedicados.

No primeiro caso, os recursos são todos idênticos e a tarefa pode escolher quaisquer recursos, desde que atenda a quantidade necessária para ser executada e algumas outras eventuais restrições. No segundo caso, os recursos são específicos e dedicados, o que significa que a tarefa precisa especificamente de um dado conjunto de recursos ao mesmo tempo para ser executada.

Nahmias (2008) ainda destaca outros problemas de programação, baseado no critério da aplicação. Além dos já descritos, Nahmias (2008) cita os problemas de programação de pessoal, como tripulações ou funcionários, programação de facilidades e programação de veículos. Essas aplicações possuem características específicas, que as diferenciam dos problemas de programação de máquinas e programação de projetos.

### 3.3 Programação de *job shop*

O problema estudado nesse trabalho pode ser abordado, em parte, como um problema de programação de *job shop*. Conforme apresentado na seção 3.2, os problemas de *job shop* são aqueles que existem diversas máquinas no sistema produtivo e as tarefas utilizam vários roteiros de produção, previamente conhecidos, para serem processadas. Os *job shops* têm como característica o fluxo de produção complexo, ou seja, cada tarefa tem a sua própria seqüência de máquinas para ser processada e essa seqüência é diferente para cada tarefa. Não somente a seqüência como o número de vezes que a tarefa é processada em uma máquina em específico pode variar também.

Considere uma máquina  $M$  que faz parte de um *job shop*. Esta máquina pode receber tarefas que foram processadas por outras máquinas, ou receber uma nova tarefa e processá-la. A tarefa pode então terminar ou seguir para processamento em outra máquina, conforme o fluxo de produção. A figura 14 ilustra os possíveis fluxos de trabalho típicos para um *job shop*.

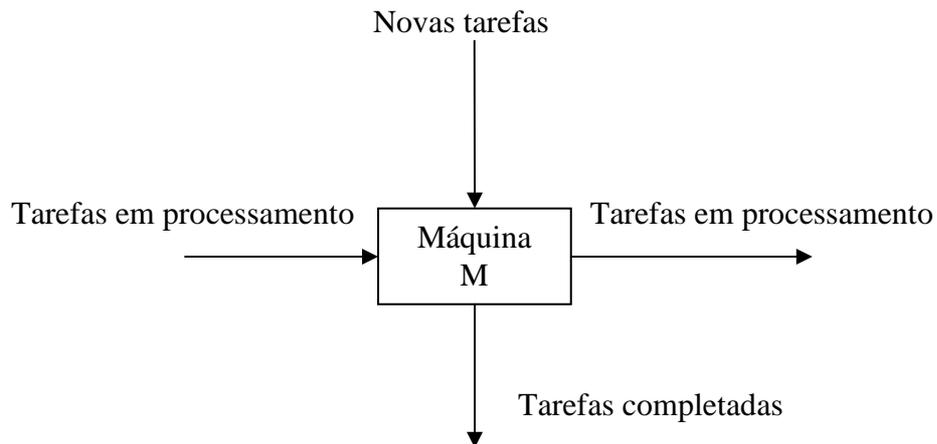


Figura 14: Fluxo de trabalho de um *job shop* (traduzido de BAKER, 1974).

Baker (1974), Coffman (1967) e Morton e Pentico (1993) apresentam em detalhes a formulação de um *job shop*. As tarefas são descritas com 3 índices,  $i$ ,  $j$  e  $k$ . O índice  $i$  identifica a tarefa, o índice  $j$  identifica a operação e o índice  $k$  identifica a máquina que a operação deve ser executada.

Considere um exemplo com 4 tarefas e 4 máquinas. Os roteiros de produção são apresentados na tabela 4. Cada linha é uma tarefa e cada coluna uma operação. Deste modo, a tarefa 1 tem a primeira operação, a operação 1 executada na máquina 1, a operação 2 executada na máquina 3 e a operação 3 executada na máquina 2. Assim pode ser representado para as outras tarefas, sucessivamente.

Tabela 2: Roteiros de produção das tarefas

Tarefa	Operação		
	1	2	3
1	1	3	2
2	3	2	1
3	4	3	-
4	1	3	4

Deste modo, as tarefas podem ser alocadas ao longo do tempo, formando um programa de produção como o apresentado na figura 15.

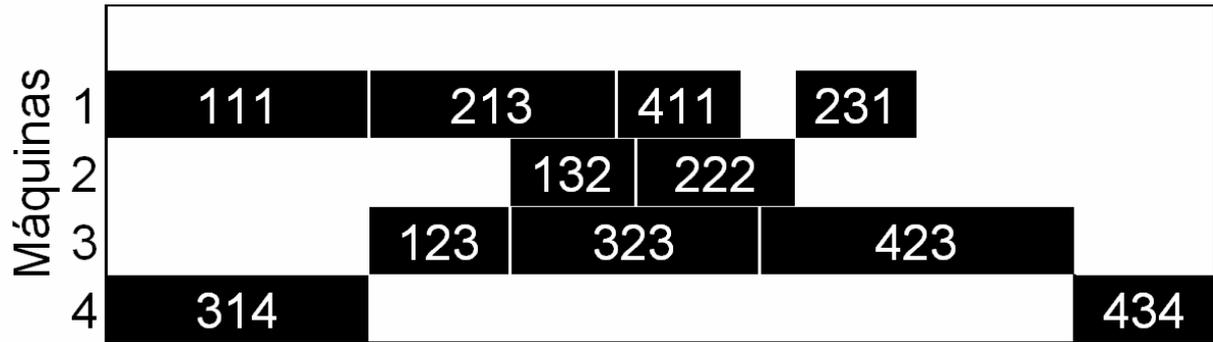


Figura 15: Exemplo de programa de produção

Esse programa foi feito considerando que existe precedência entre as operações. Outras soluções viáveis para o mesmo conjunto de tarefas podem ser encontradas. Para um *job shop* formado por  $n$  tarefas, com uma média de  $q$  operações por tarefa e  $m$  máquinas, o número de combinações possíveis é dado por:

$$\left[ \left( \frac{n \cdot q}{m} \right)! \right]^m \quad (1)$$

Se o sistema é um *job shop* com 4 tarefas, 5 operações por tarefa e 4 máquinas, pela expressão (1), o número de possíveis programações é de 207.360.000. Esse número de combinações poderia ser reduzido devido aos roteiros de produção, que restringiriam o número de máquinas possíveis de ser utilizadas por cada operação. Mas, mesmo assim, sistemas de *job shop* reais possuem, em geral, um número bem maior de tarefas, operações e máquinas. Por isso, a utilização de métodos matemáticos é necessária para resolver esses problemas e, dependendo do caso, apenas soluções heurísticas são possíveis.

Baker (1974) propõe três abordagens para resolver o problema de programação de *job shop*. A primeira é a utilização do método *branch-and-bound*. A segunda é a utilização de heurísticas específicas e a terceira é o uso de modelos matemáticos lineares mistos (WOLSEY, 1998; GOLDBARG; LUNA, 2005).

Uma formulação de modelo de programação inteira possível para o problema de *job shop* é o modelo que usa restrições disjuntivas (ARENALES et al., 2006). Considere  $x_{ik}$  como a variável de decisão para o momento de término da tarefa  $i$  na máquina  $k$ . Assim, o conjunto de valores das variáveis  $x_{ik}$  representa um programa de produção e é estritamente positiva. Suponha que a operação  $j$  da tarefa  $i$  requer a máquina  $k$  e a operação  $(j-1)$  da tarefa  $i$  requer a máquina  $h$ . Então:

$$x_{ik} - t_{ijk} \geq x_{ih} \quad (2)$$

em que  $t_{ijk}$  é a duração da operação  $j$  na máquina  $k$  da tarefa  $i$ . A restrição (2) garante que a operação  $(j-1)$  da tarefa  $i$  termine no mínimo  $t_{ijk}$  unidades de tempo antes do término da operação  $j$  da mesma tarefa  $i$  ( $x_{ik}$ ), ou seja, a restrição (2) garante o respeito às relações de precedência das operações  $j$  e  $j-1$  de uma tarefa  $i$ .

Se as tarefas  $i$  e  $p$  têm as operações  $q$  e  $j$ , respectivamente, executadas consecutivamente na máquina  $k$ , então uma das duas restrições a seguir é válida:

$$x_{pk} - t_{pqk} \geq x_{ik} \quad (3)$$

$$x_{ik} - t_{ijk} \geq x_{pk} \quad (4)$$

A restrição (3) precisa ser válida somente se a tarefa  $i$  precede a tarefa  $p$ . A restrição (4) precisa ser válida somente se a tarefa  $p$  precede a tarefa  $i$  na máquina  $k$ . Por isso, é necessário utilizar as restrições (3) e (4) de forma disjuntiva, ou seja, quando uma está ativa, a outra é desativada e vice-versa. Uma maneira de fazer isso é utilizar uma variável binária e um número positivo suficientemente grande,  $H$ . Então, as restrições (3) e (4) podem ser reescritas como:

$$x_{pk} - x_{ik} + H \cdot (1 - y_{ipk}) \geq t_{pqk} \quad (5)$$

$$x_{ik} - x_{pk} + H \cdot y_{ipk} \geq t_{ijk} \quad (6)$$

em que  $y_{ipk}$  é 1 se a tarefa  $i$  precede a tarefa  $p$  na máquina  $k$  e 0 caso contrário.

A função objetivo depende da decisão envolvida, podendo ser minimizar o tempo para completar todas as tarefas, o *makespan*, o tempo médio de fluxo, ou um outro objetivo coerente com o problema estudado. A seguir, é apresentado o modelo clássico de *job shop* com o objetivo de minimizar o *makespan*. Considere:

$C_{makespan}$ : tempo de término da última tarefa

Então o modelo é:

$$\text{Min } C_{makespan} \quad (7)$$

$$C_{makespan} \geq x_{ik} \quad \forall i, k \quad (8)$$

$$x_{ik} \geq t_{ijk} \quad \forall i, k; j = 1 \quad (9)$$

$$x_{ik} - t_{ijk} \geq x_{ih} \quad \forall i, k \text{ se } h \rightarrow k \quad (10)$$

$$x_{pk} - x_{ik} + H \cdot (1 - y_{ipk}) \geq t_{pqk} \quad \forall p, i, k \quad (11)$$

$$x_{ik} - x_{pk} + H \cdot y_{ipk} \geq t_{ijk} \quad \forall p, i, k \quad (12)$$

$$x_{ik} \in R^+, y_{ipk} \in \{0, 1\} \quad (13)$$

A função objetivo é minimizar o *makespan* (7). As restrições (8) garantem que o *makespan* seja igual ao momento de término da última tarefa do programa. A restrição (9) garante que a primeira operação termine pelo menos o tempo que demora a sua execução. As restrições (10) garantem as relações de precedência entre as operações. As restrições (11) e (12) são as restrições disjuntivas que impedem que duas tarefas que utilizam a mesma máquina sejam executadas ao mesmo tempo. A restrição (13) é referente ao domínio das variáveis.

O modelo apresentado poderia ser implementado em *software* de programação matemática específico como o GAMS (*Generic Algebraic Modelling System*) e resolvido com um *software* apropriado como o CPLEX. Este modelo considera que os tempos de processamento são conhecidos e constantes, ou seja, caso determinístico, e todas as tarefas estão disponíveis para serem processadas no instante inicial do programa, ou seja, o processo de chegada das tarefas é estático.

### 3.4 Programação de projetos

A montagem em gabarito estudada nesse trabalho também pode ser abordada como um problema de programação de projetos, conforme é apresentado na seção 4. A programação de projetos tem a sua origem com o desenvolvimento de duas técnicas, criadas na década de 50, responsáveis pela formalização da programação de projetos. As técnicas são o CPM (*critical path method*) e o PERT (*project evaluation and review technique*). Nessas técnicas, as tarefas a serem executadas, os recursos que elas utilizam e as suas relações entre si são apresentadas na forma de uma rede. Nahmias (2008) apresenta as duas técnicas e observa que o uso de redes para representar projetos é mais adequado do que o uso apenas de gráficos de Gantt, pois as redes conseguem representar explicitamente as relações de precedência das tarefas e os gráficos de Gantt não.

A literatura de programação de projetos costuma denominar o conjunto mais elementar de ações de produção como atividade e um conjunto de atividades formam uma tarefa. Um conjunto de tarefas forma um projeto. Uma rede é um conjunto de pontos, chamados nós, e um conjunto de arcos, relacionando os nós. Existem duas formas de representar um projeto em rede; as atividades de um projeto podem ser representadas nos arcos ou nos nós de uma rede. Morton e Pentico (1993) afirmam que a representação das atividades nos nós foi mais amplamente utilizada, mas a representação em arcos tem sido

mais popularmente usada nos últimos anos. A figura 16 apresenta as duas formas de representação do projeto em rede.

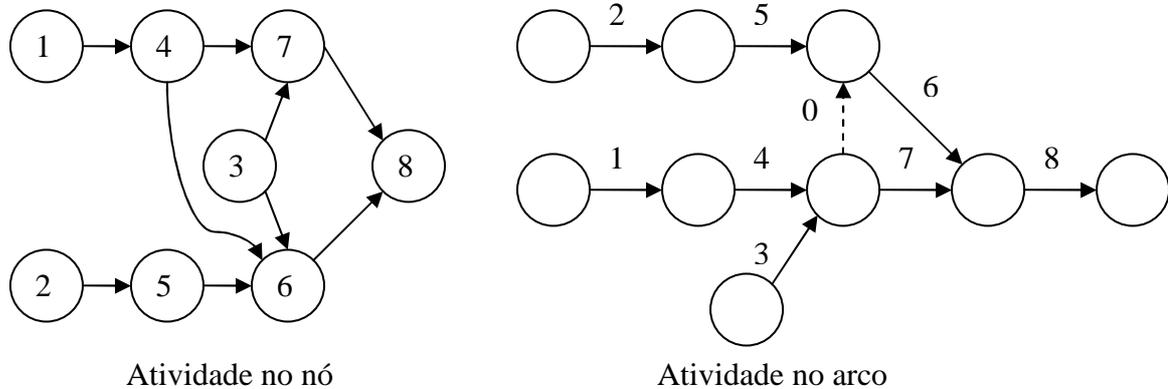


Figura 16: Formas de representação da rede de relacionamento das tarefas de um projeto (PINEDO, 2005).

Pinedo (2005) afirma que a desvantagem da representação da atividade no arco é a necessidade de utilizar atividades fantasmas (*dummy*) para manter as relações de precedência. As atividades fantasmas são atividades que não consomem recursos e têm duração nula. Segundo Pinedo (2005), o número de atividades fantasmas pode chegar a 10% em um projeto real, aumentando significativamente o número total de atividades. Outra vantagem da representação das atividades nos nós é que se pode utilizar retângulos para representar as atividades, inclusive em escala, e assim, utilizar as vantagens do gráfico de Gantt também.

Na representação de atividade no arco, o término de uma atividade pode, e geralmente é desejável, coincidir com início da próxima atividade. Morton e Pentico (1993) afirmam que existem algumas regras prescritas para a utilização da representação da atividade no arco. A rede deve ter um único nó de início e um único nó de término, cada atividade só pode ser representada por um arco, e duas atividades não possuem um mesmo evento de início e término. Como cada atividade só pode ser representada por um único arco, a atividade 3 da figura 16, que precede as atividades 7 e 6, necessita ser representada por um arco que precede a atividade 7 e um outro arco fantasma, para garantir a precedência da atividade 6.

Morton e Pentico (1993) ainda afirmam que, para finalidades informais ou estimativas manuais, a última regra não precisa ser respeitada com rigor. No entanto, em projetos grandes, utilizar o mesmo evento de início e término para atividades diferentes é uma ambigüidade que deve ser evitada. A figura 17 ilustra um caso em que isso ocorre. As atividades B e C têm os mesmos eventos de início e término, 3 e 4, conforme a figura 17 (a), mas a representação correta em redes de projeto é a figura 17 (b). Nesse caso, as tarefas B e C

têm o mesmo evento de início, o evento 2, mas os eventos de término são diferentes. Para manter a relação entre as tarefas, é utilizada uma tarefa fantasma, representada pelo arco tracejado da figura 17 (b).

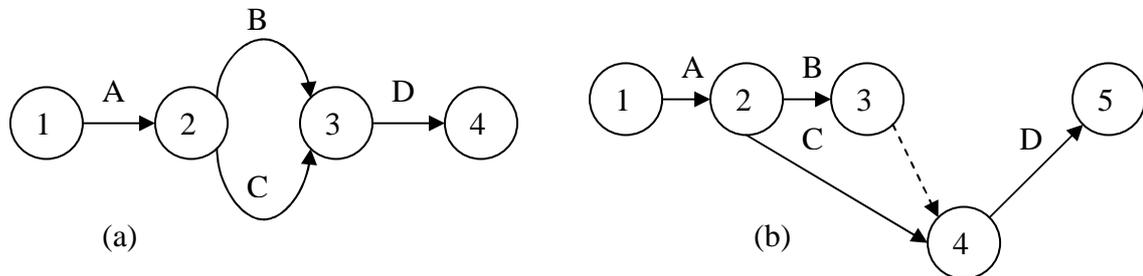


Figura 17: (a) Representação informal das atividades nos arcos (b) Representação formal das atividades nos arcos (MORTON; PENTICO, 1993).

Uma outra regra que também é utilizada na representação de atividades em arcos é sempre representar um evento de término de uma atividade por um número maior do que o evento de início. Essa regra é válida apenas para o caso em que se opta por representar as atividades como letras e os eventos como números. Foi apenas nos anos 80 que a programação de projetos foi abordada por modelos matemáticos e, ainda hoje, possuem grandes perspectivas de pesquisa e aplicações em problemas práticos. Alguns autores como Morton e Pentico (1993) e Brucker et al. (1999) apresentam os problemas de programação de *job shop* como um caso particular dos problemas de programação de projetos. Frequentemente, os projetos têm suas atividades com tempos que podem variar conforme a quantidade de recursos que o responsável pelo projeto aloca a cada atividade. Em construção civil, por exemplo, se uma atividade é intensiva em mão de obra, a alocação de maior quantidade de pessoas à atividade pode diminuir o tempo necessário para executar a atividade. A redução do tempo em função da quantidade de mão de obra pode ser linear ou não.

Na execução de um projeto, os responsáveis pelo projeto frequentemente possuem uma quantidade limitada de recursos, geralmente um orçamento, e um conjunto de atividades que podem variar a quantidade de recursos e assim, a velocidade de execução. Uma questão fundamental em administração de projetos é quanto alocar de recursos a cada atividade, de forma a atender todos os requisitos contratados como prazo de entrega e qualidade, dado que a quantidade de recursos disponíveis é limitada. Essa questão está justamente no conhecido *trade-off* entre quantidade de recursos utilizados para execução do projeto e custo total do projeto que inclui, por exemplo, custos do eventual não atendimento dos prazos contratados, ou simplesmente *trade-off* entre tempo e custo.

Os problemas que possuem restrição de quantidade de recursos são conhecidos como problemas de programação de projetos com restrição de recursos (RCPSP – *resource-constrained project scheduling problem*). Nos casos em que se pode executar as atividades em diferentes velocidades conforme a quantidade de recursos alocados, a literatura chama esses casos de atividades com vários modos de execução. Brucker et al. (1999) apresenta uma classificação e uma notação para esses dois tipos de problemas. Além disso, Brucker et al. (1999) e Kolish e Padman (2001) fazem uma extensa revisão dos modelos matemáticos e algoritmos exatos e heurísticos que resolvem as diversas variações dos problemas de programação de projetos, como funções objetivo, presença ou não de recursos renováveis, e tempos de processamento determinísticos ou estocásticos.

Os recursos renováveis dentro do contexto de programação de projetos, segundo Brucker et al. (1999), são todos aqueles recursos que, depois de utilizados, não diminuem a quantidade disponível. Tipicamente, a mão de obra é um recurso renovável. Os recursos não renováveis são aqueles que, depois de consumidos, não podem ser utilizados novamente. Em um projeto de construção civil, o cimento utilizado em uma atividade não é um recurso renovável.

Kolish e Padman (2001) ainda afirmam que existe um terceiro tipo de recurso, o parcialmente renovável. São aqueles recursos que tem a utilização restrita a um subconjunto de tempo dentro do horizonte de tempo considerado. Kolish e Padman (2001) citam como exemplo o uso de mão de obra que, dentro de um horizonte de tempo mensal, tem sua disponibilidade restrita por força de contrato aos dias úteis. Kolish e Padman (2001) também apresentam os problemas de programação de projetos que consideram o fluxo de caixa na execução das tarefas e como as duas representações das redes, atividades nos nós e atividades nos arcos, impactam na representação do fluxo de caixa do projeto.

Pinedo (2005) apresenta o modelo clássico de programação de projetos com restrição de recursos e tempos de processamento determinísticos. Considere:

$p_j$  : tempo de processamento da atividade  $j$

$N$  : número de diferentes tipos de operadores

$W_i$  : número total de operadores do tipo  $i$

$W_{ij}$  : é o número de operadores do tipo  $i$  que a atividade  $j$  necessita

$A$  : conjunto dos pares de atividades  $j$  e  $k$ , tal que  $j$  precede  $k$  ( $j \rightarrow k$ )

As variáveis de decisão são:

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

O número de operadores do tipo  $i$  que a atividade  $j$  utiliza no intervalo  $[t - 1, t]$  é dado pela expressão (14).

$$W_{ij} \cdot \sum_{u=t}^{t+p_j-1} x_{ju} \quad (14)$$

Como se pode perceber, o modelo considera o tempo de forma discreta. É necessário um limitante superior para o *makespan*, denominado  $H$ . Um limitante superior simples é a soma dos tempos de processamento de todas as atividades, como na expressão (15).

$$H = \sum_{j=1}^n p_j \quad (15)$$

Considerando que  $H$  é inteiro, então, o instante de término de cada atividade  $j$  é dado pela expressão (16).

$$\sum_{t=1}^H t \cdot x_{jt} \quad (16)$$

Como na representação de atividades nos arcos, é necessária a utilização de uma atividade que deve ser a última a ser executada, mas tem duração nula e não consome nenhum recurso. Essa atividade é chamada de atividade  $n + 1$ . Assim, o tempo de término de todas as atividades é o tempo de término da atividade  $n + 1$ , ou seja, o *makespan*, conforme a expressão (17).

$$\sum_{t=1}^H t \cdot x_{n+1,t} \quad (17)$$

Assim, Pinedo (2005) apresenta o modelo formulado como:

$$\text{Minimizar } \sum_{t=1}^H t \cdot x_{n+1,t} \quad (18)$$

Sujeito a:

$$\sum_{t=1}^H t \cdot x_{jt} + p_k - \sum_{t=1}^H t \cdot x_{kt} \leq 0 \quad \text{Para todo } j \rightarrow k \in A(j, k) \quad (19)$$

$$\sum_{j=1}^n \left( W_{ij} \cdot \sum_{u=t}^{t+p_j-1} x_{ju} \right) \leq W_i \quad \text{para todo } i \text{ e } t \quad (20)$$

$$\sum_{t=1}^H x_{jt} = 1 \quad \text{para todo } j \quad (21)$$

$$x_{jt} \in \{0, 1\} \quad (22)$$

A função objetivo (18) é minimizar o *makespan*. As restrições (19) garantem o respeito às relações de precedência entre as tarefas. As restrições (20) garantem que a quantidade do recurso  $i$  em uso a cada momento  $t$  para executar todas as atividades respeite a quantidade total disponível. As restrições (21) garantem que todas as atividades serão programadas. As restrições (22) são referentes ao domínio das variáveis. Nesse modelo clássico, não existem vários modos de execução, mas, cada atividade pode precisar de diversos recursos ao mesmo tempo para ser executada. No caso do modelo (18)-(22), o recurso utilizado é a mão de obra,  $W_{ij}$ , mas poderiam ser quaisquer outros recursos renováveis.

Carlier et al. (2009) fazem um estudo dos algoritmos desenvolvidos para o caso específico que o projeto consome recursos não renováveis. Guldmond et al. (2008) apresentam um estudo para o problema de programação de projetos com restrição de tempo (TCPSP – *time constrained project scheduling problem*), que é uma variação do RCPSP. No caso do TCPSP, o tempo para terminar o projeto é estritamente definido e deve-se então, alocar a quantidade de recursos necessária para completar o projeto naquele período. Para cumprir o prazo, pode-se subcontratar algumas atividades ou utilizar horas extras e, então, o problema é definir quais recursos, quantos e quando serão necessários para cumprir esse prazo. No RCPSP, minimiza-se o *makespan*, para uma quantidade fixa de recursos disponíveis.

Considere os mesmos parâmetros apresentados no modelo anterior de programação de projetos. Considere também:

$r$  : tipo de recurso,  $r = 1, \dots, R$

$T$  : prazo máximo para entrega de todas as atividades

$r_{jr}$  : quantidade do recurso  $r$  necessária para executar a atividade  $j$

As variáveis de decisão são:

$a_r$  : quantidade a ser disponibilizada do recurso  $r$

$v_r$  : custo de disponibilizar uma unidade do recurso  $r$

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

Então, um modelo matemático de TCPSP é:

$$\text{Min } \sum_{r=1}^R v_r \cdot a_r \quad (23)$$

$$\sum_{t=1}^T t \cdot x_{jt} + p_k - \sum_{t=1}^T t \cdot x_{kt} \leq 0 \quad \text{para todo } j \rightarrow k \in A(j, k) \quad (24)$$

$$\sum_{j=1}^J \sum_{b=t}^{t+p_j-1} r_{jr} \cdot x_{jb} \leq a_r \quad \forall t, r \quad (25)$$

$$\sum_{t=1}^T x_{jt} = 1 \quad \forall j \quad (26)$$

$$x_{jt} \in \{0, 1\}; a_r \in \mathbb{R}; a_r \geq 0; t = 1, \dots, T; r = 1, \dots, R; j = 1, \dots, J \quad (27)$$

A função objetivo (23) é minimizar o custo total dos recursos a ser disponibilizado para executar o projeto. As restrições (24) garantem o respeito às relações de precedência. As restrições (25) garantem que a quantidade de recursos utilizados em qualquer momento  $t$  não ultrapasse a quantidade a ser disponibilizada,  $a_r$ . As restrições (26) garantem que cada tarefa é programada uma e apenas única vez. As restrições (27) são referentes ao domínio das variáveis.

A função objetivo poderia ter associada um custo para cada recurso utilizado ou ainda, outras variáveis representando recursos diferentes como horas extras e subcontratação, e o objetivo seria minimizar o custo total, caracterizando o problema de programação de projetos com custos de disponibilidade de recursos (*resource availability cost problem*). Nesse caso, a duração das tarefas,  $p_j$ , poderia variar conforme as quantidades de recursos utilizadas, ou seja, a mesma tarefa pode ter vários modos de execução. Yamashita et al. (2006, 2007) e Yamashita e Morabito (2007a, 2007b, 2009) apresentam um estudo do problema de programar projetos com múltiplos modos de execução e custos de disponibilização de recursos, com objetivo de minimizar o *trade-off* de custo e tempo. Herroelen e Leus (2005) fazem uma extensa revisão das pesquisas e métodos de resolução dos problemas de programação de projetos, porém, considerando incertezas em diversos aspectos como a duração ou a interrupção das tarefas.

Um modelo de programação de projetos com restrição de tempo e múltiplos modos (MMTCPSP – *multi-mode time constrained project scheduling problem*) é apresentado a seguir, baseado em Yamashita e Morabito (2007a). Considere os seguintes parâmetros:

$i$  : modo de execução,  $i = 1, \dots, I$

$t$  : período de tempo,  $t = 1, \dots, T$

$r$  : tipo de recurso,  $r = 1, \dots, R$

$j$  : atividade,  $j = 1, \dots, J$

$p_{ji}$  : tempo de processamento da atividade  $j$  no modo de execução  $i$

$A$  : conjunto dos pares de atividades  $j$  e  $k$ , tal que  $j$  precede  $k$  ( $j \rightarrow k$ )

$r_{jir}$  : quantidade do recurso  $r$  utilizada para executar a tarefa  $j$  no modo  $i$

As variáveis de decisão são:

$a_r$  : quantidade a ser disponibilizada do recurso  $r$

$v_r$  : custo de disponibilizar uma unidade do recurso  $r$

$x_{jti} = \begin{cases} 1 & \text{se a atividade } j \text{ termina exatamente no tempo } t \text{ no modo de execução } i \\ 0, & \text{caso contrário} \end{cases}$

$$\text{Min } \sum_{r=1}^R v_r \cdot a_r \quad (28)$$

$$\sum_{i=1}^I \sum_{t=1}^T t \cdot x_{jti} + p_{ki} - \sum_{i=1}^I \sum_{t=1}^T t \cdot x_{kti} \leq 0 \quad \text{Para todo } j \rightarrow k \in A (j, k) \quad (29)$$

$$\sum_{j=1}^J \sum_{i=1}^I \sum_{b=t}^{t+p_{ji}-1} r_{jir} \cdot x_{jbi} \leq a_r \quad \forall t, r \quad (30)$$

$$\sum_{i=1}^I \sum_{t=1}^T x_{jti} = 1 \quad \forall j \quad (31)$$

$$x_{jti} \in \{0, 1\}; a_r \in \mathbb{R}; a_r \geq 0; t = 1, \dots, T; r = 1, \dots, R; j = 1, \dots, J \quad (32)$$

A função objetivo (28) minimiza o custo total dos recursos a serem utilizados no projeto. As restrições (29) garantem o respeito às relações de precedência. As restrições (30) garantem a utilização de apenas a quantidade de recursos disponível. As restrições (31) garantem que cada tarefa é alocada uma e apenas uma única vez para todo o horizonte de tempo. As restrições (32) são referentes ao domínio das variáveis.

### 3.5 Restrições de adjacência

O desenvolvimento de novas tecnologias e processos de produção cria situações e problemas de produção novos e que, muitas vezes, ainda não foram estudados pela pesquisa operacional ou pela programação da produção. A programação da produção com restrições de adjacência surgiu da aplicação dos conceitos de programação em, por exemplo,

planejamento de colheitas de madeira ou controle de semáforos, mas também do desenvolvimento de novas tecnologias de produção, como os gabaritos de montagem na indústria aeronáutica ou os novos sistemas de comunicação via celular e mais recentemente, redes *wireless*.

Os problemas de programação com restrições de adjacência são uma classe de problemas na literatura de programação de produção, com tipologia e formulação ainda não claramente definidas. Por isso, esses problemas são atualmente uma perspectiva de pesquisa em programação da produção. Estes problemas surgem nas publicações com diversos nomes diferentes, como programação de tarefas de exclusão mútua (*mutual exclusion scheduling*), em Baker e Coffman (1996) e Jansen (1999; 2003), programação com conflitos (*scheduling with conflicts*) em Irani e Leung (1996; 2003), Kaplan (2007) e Even et al. (2009), programação com restrição de adjacência (*adjacent constraint scheduling*) em Santos et al. (2007; 2008), e Gómez et al. (2009) ou programação de recursos adjacentes em Duin e Sluis (2006).

No presente trabalho, o nome adotado para essa classe de problemas é problema de programação da produção com restrições de adjacência. Embora os problemas apresentados nesta revisão possuam várias características particulares, todos têm uma característica em comum: a restrição de adjacência. Os problemas de programação com restrições de adjacência são definidos como: dado um recurso qualquer que possui a capacidade de processar múltiplas tarefas ao mesmo tempo (*multiprocessor*) em várias estações de trabalho idênticas ou não, a programação com restrição de adjacência surge quando a programação de uma tarefa  $j$  interfere na possibilidade de execução de uma tarefa  $k$ , sendo que  $j$  e  $k$  não utilizam as mesmas estações de trabalho.

A tarefa  $j$  pode interferir na execução da tarefa  $k$  ocupando ou bloqueando um recurso que é necessário para a execução de  $k$ . Considere três tarefas  $i$ ,  $j$  e  $k$ , e três recursos,  $r_1$ ,  $r_2$  e  $r_3$ . A tarefa  $i$  utiliza o recurso  $r_1$ , a tarefa  $j$  utiliza o recurso  $r_2$  e a tarefa  $k$  utiliza o  $r_3$ . Mas durante a execução da tarefa  $i$ , o recurso  $r_2$  fica bloqueado, por exemplo, por restrição de espaço. Por  $r_2$  estar bloqueado, a tarefa  $j$  não pode ser executada ao mesmo tempo em que a tarefa  $i$  é executada. Supondo que a tarefa  $k$ , que utiliza  $r_3$  também bloqueie  $r_2$ , então, as tarefas  $j$  e  $k$  também não podem ser executadas ao mesmo tempo. No entanto, as tarefas  $i$  e  $k$ , apesar de ambas bloquearem  $r_2$ , não utilizam  $r_2$  e podem ser executadas ao mesmo tempo. A figura 18 ilustra a relação entre as tarefas  $i$ ,  $j$  e  $k$  e um exemplo de programa viável para produção das tarefas. Perceba no diagrama de Venn que as tarefas  $i$  e  $k$  podem ser executadas

ao mesmo tempo e por isso possuem uma intersecção, enquanto a tarefa  $j$  possui restrições de adjacência em relação às tarefas  $i$  e  $k$  e, assim, não possui intersecção com estas.

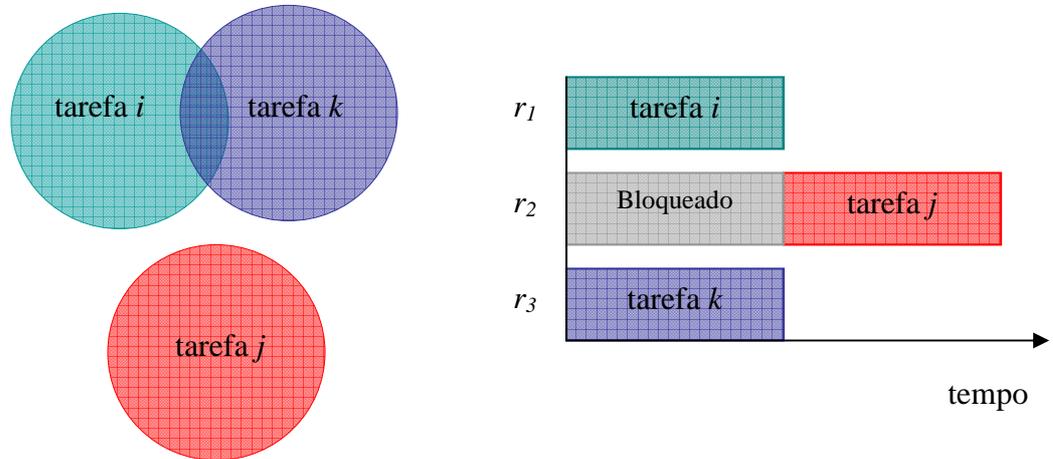


Figura 18: Tarefas que bloqueiam os recursos e um Gantt de um programa ilustrativo.

Considere agora que a tarefa  $i$  precisa de  $r_1$  e  $r_2$  ao mesmo tempo, para ser executada. A tarefa  $j$  utiliza apenas  $r_2$  e a tarefa  $k$  utiliza apenas  $r_3$ . Então, a tarefa  $i$  impede a execução da tarefa  $j$ , mas não impede a execução da tarefa  $k$ . Da mesma forma,  $j$  impede a execução da tarefa  $i$ , mas não de  $k$ . A figura 19 ilustra o diagrama de intersecção das tarefas e um programa exemplo no gráfico de Gantt. Perceba pelo diagrama de Venn que as tarefas  $i$  e  $k$  e  $k$  e  $j$  possuem intersecções entre si, o que significa que podem ser executadas ao mesmo tempo, mas as tarefas  $j$  e  $i$  não possuem intersecção, pois não podem ser executadas ao mesmo tempo devido às restrições de adjacência.

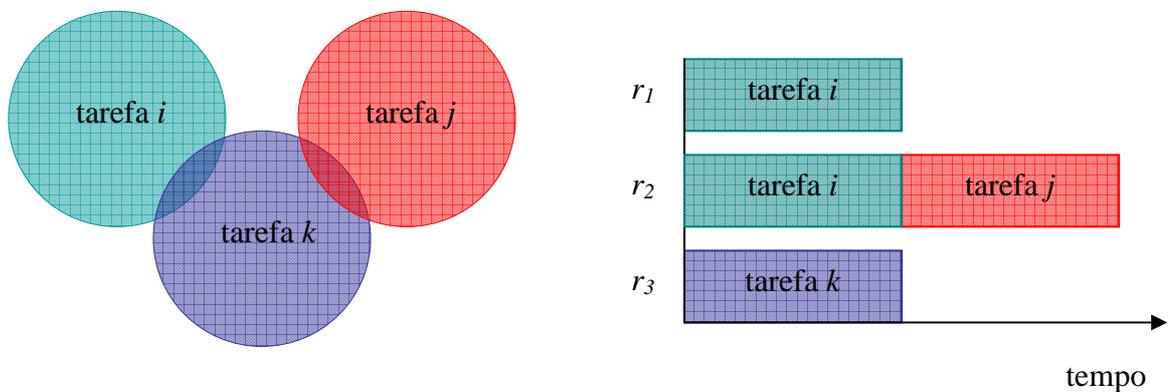


Figura 19: Tarefas que utilizam os recursos e um Gantt de um programa ilustrativo.

As relações de bloquear ou ocupar os vários recursos são as restrições de adjacência que podem ser representadas por um grafo, em que cada nó é uma tarefa e as arestas ligam os pares de tarefas que possuem restrições de adjacência. A figura 20 ilustra os grafos de conflito da figura 18 e da figura 19.

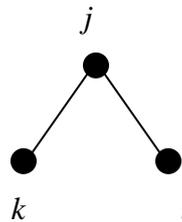


Figura 19

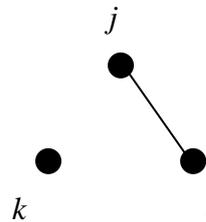


Figura 20

Figura 20: Grafo de conflito dos exemplos apresentados na figura 19 e figura 20

Na literatura clássica de programação da produção (e.g., Conway et al., 1967; Baker, 1974; Coffman et al., 1976; Hax; Candeia, 1984; Morton; Pentico, 1993; Nahmias, 2008, Pinedo, 2005, 2008), esse problema não aparece formalmente definido. Os problemas de programação com restrições de adjacência aparecem em diversos campos de aplicação da pesquisa operacional, mas aparentemente não fazem referências umas às outras. Em pesquisa realizada na base *web of knowledge*, foram identificadas cinco áreas em que as restrições de adjacência são utilizadas: o planejamento de colheitas de florestas e rotação de culturas na agricultura, nas telecomunicações, em controle de tráfego de veículos, programação de projetos com restrições de recursos e *bin packing* com conflitos.

O planejamento de florestas e agricultura aborda os três diferentes horizontes de planejamento, o estratégico, tático e operacional. A restrição de adjacência surge nos problemas de planejamento tático e operacional de colheitas de florestas. As decisões envolvidas nesses problemas de planejamento variam muito conforme as florestas, leis ambientais, tipo de cultura e objetivo a ser otimizado. Genericamente, no planejamento de florestas, deve-se programar a colheita de árvores em áreas determinadas, decidindo onde, quanto e quando serão extraídas as madeiras, maximizando os retornos financeiros ou minimizando custos, sujeito às restrições relativas à localização de equipamentos de colheita, transporte e ambientais relativos tanto à fauna quanto à flora.

Goméz et al. (2009) afirmam que as restrições de adjacência já são bem conhecidas na literatura de planejamento de colheita em florestas. Elas servem para preservar a beleza visual e a biodiversidade da floresta em exploração, e evitam a erosão. Weintraub et

al. (2007) acrescentam que a programação de colheitas no planejamento de florestas permite manter a qualidade da água, promover a diversidade e a sustentabilidade de modo geral.

Dada uma área de floresta, esta é dividida em áreas de manejo com tamanho regulado por agências de governo de cada país, sendo tipicamente utilizado entre 25 e 49 ha como tamanho máximo de cada área (GOYCOOLEA et al., 2005). Assim, a floresta fica dividida em áreas de manejo como as apresentadas na figura 21. As áreas de manejo adjacentes são aquelas que dividem alguma fronteira em comum ou possuem certa distância máxima. Assim, não se pode cortar duas áreas adjacentes, em um período  $t$  qualquer ou em  $p$  períodos depois do corte no período  $t$ , a fim de garantir a regeneração da área de manejo, chamado *green-up* (GOYCOOLEA et al., 2005).

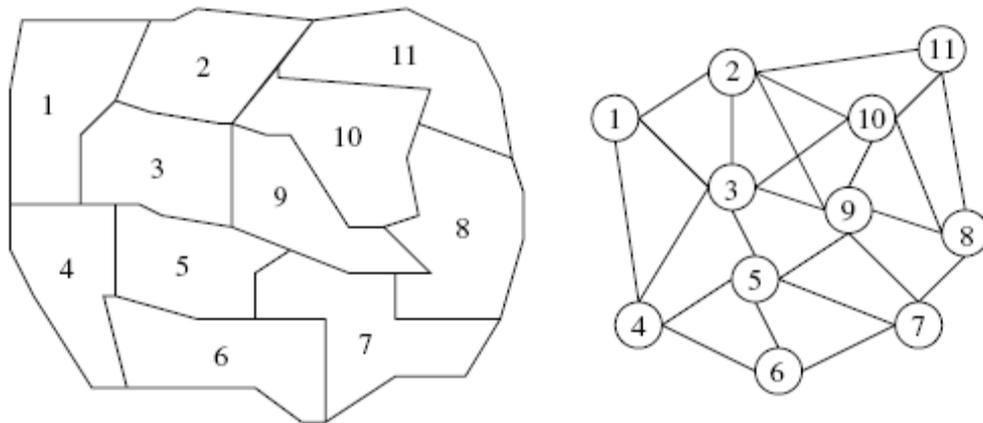


Figura 21: Floresta dividida em áreas de manejo e o respectivo grafo representando as relações de adjacência das áreas de manejo (GOYCOOLEA et al., 2005).

A modelagem matemática desse problema pode utilizar uma formulação binária e utilizar diversas técnicas de solução, como *branch-and-bound*, algoritmos genéticos ou busca tabu, já que o problema é NP - difícil e problemas práticos podem ser difíceis de resolver com modelos matemáticos (GOYCOOLEA et al., 2005). Duas formulações clássicas do problema de programação de colheitas com restrições de adjacência são reportadas na literatura. A primeira é um modelo de restrição de unidade (*unit restriction model* – URM) e a segunda é um modelo de restrição de área (*area restriction model* – ARM). A diferença entre eles é que o URM considera a adjacência entre as áreas de manejo, mas não considera o tamanho das áreas. O ARM considera além da adjacência, o tamanho das áreas de manejo que são programadas para a colheita. Weintraub et al. (2007) apresentam em detalhes as considerações dos modelos e demonstram que o URM é um caso particular do ARM. Então, a seguir é apresentado apenas o modelo ARM. A formulação clássica do ARM, apresentada a seguir, é apresentada por Weintraub et al. (2007). Considere os seguintes parâmetros:

$t$  : período de tempo

$l$  : área de manejo

$p$  : período de colheita proibida

$\beta_{lt}$  : é o benefício de colher a área de manejo  $l$  no período  $t$ ,

$\Omega_l$  : é o conjunto de áreas adjacentes à área de manejo  $l$ ,

$v_{lt}$  : é o volume produzido pela colheita no bloco  $l$  no período  $t$ ,

$U_t$  : é o limite superior do volume total a ser colhido no período  $t$

$L_t$  : é o limite inferior do volume total a ser colhido no período  $t$ .

A variável de decisão é  $y_{lt}$ , em que:

$$y_{lt} : \begin{cases} 1 & \text{se a colheita na área } l \text{ é feita no período } t \\ 0 & \text{caso contrário} \end{cases}$$

Então, o modelo é:

$$\text{Maximizar } \sum_l \sum_t \beta_{lt} \cdot y_{lt} \quad (33)$$

$$\sum_{t'=t-p}^{t+p} (y_{lt'} + y_{l't'}) \leq 1 \quad \forall l, l' \in \Omega_l, t \quad (34)$$

$$\sum_l v_{lt} y_{lt} \leq U_t \quad \forall t \quad (35)$$

$$\sum_l v_{lt} y_{lt} \geq L_t \quad \forall l \quad (36)$$

$$\sum_t y_{lt} \leq 1 \quad \forall l \quad (37)$$

$$y_{lt} \in \{0, 1\} \quad \forall l, t \quad (38)$$

O objetivo (33) é maximizar o retorno. A restrição (34) é a restrição que garante que duas áreas de manejo adjacentes não sejam colhidas ao mesmo tempo. A restrição (35) e (36) limitam o mínimo e máximo a ser colhido. A restrição (37) garante que a área só seja utilizada uma vez dentro do horizonte de planejamento e a restrição (38) é referente ao domínio das variáveis.

Santos et al. (2007; 2008) apresentam um modelo linear inteiro binário de programação para rotação de culturas em hortas. Por questões relacionadas ao uso de nutrientes do solo e controle de pragas, uma horta é dividida em setores que não podem plantar as mesmas culturas em áreas adjacentes. Além disso, a diversificação também garante o aumento da produtividade e adaptação às variações de demanda nos produtos, entre muitas

outras vantagens (SANTOS et al., 2007). A figura 22 apresenta um exemplo de área de plantação e o respectivo grafo de conflito das subáreas adjacentes.

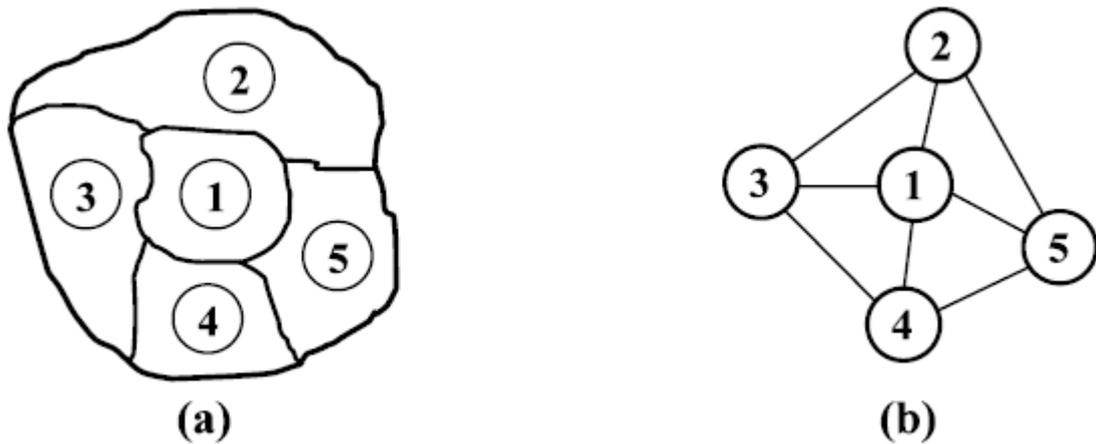


Figura 22: Plantação de hortaliças e o respectivo grafo de adjacência (SANTOS et al., 2008).

O modelo utilizado também é um modelo linear binário, com uma variável binária que representa a decisão de plantar uma cultura em uma área para um tempo  $t$ . Do ponto de vista de programação da produção, a área de floresta ou de agricultura pode ser entendida como uma máquina; cada área é dividida em setores ou, no caso de uma máquina, várias estações de trabalho; tanto os setores quanto as estações de trabalho podem ser adjacentes ou não. As tarefas a serem processadas são colheitas de árvores no caso das florestas, plantações no caso da agricultura ou peças a serem montadas no caso de uma máquina; então, as áreas de floresta de manejo, agricultura e máquinas com várias estações de trabalho são multiprocessadores, ou seja, recursos produtivos que podem processar várias tarefas ao mesmo tempo. Como os multiprocessadores possuem várias estações de trabalho, dependendo das características do problema, a execução de uma tarefa pode impedir que uma outra tarefa, que não utilize a mesma estação de trabalho, ocorra simultaneamente. Os modelos de Santos et al. (2007; 2008) e Weintraub et al. (2007) dividem o tempo em unidades de igual duração e, então, programam tarefas com iguais durações no tempo.

Outra aplicação de programação com restrições de adjacência é nas telecomunicações. Com o desenvolvimento da telefonia celular e das redes sem fio (*wireless*), os problemas de redes de telecomunicações estão cada dia mais crescendo em importância. Um dos problemas mais comuns em redes de telecomunicações é o problema de terminais ou nós escondidos (*hidden terminal problem* ou *hidden node*). Uma rede de telecomunicações é formada por um conjunto de nós que representam os transmissores e receptores da rede. Cada nó que transmite informação possui uma área limitada de cobertura, que geralmente é circular,

mas pode ser uma seção de circunferência para antenas direcionais e, em ambos os casos, o tamanho é dado pela potência de transmissão do nó.

Dado um nó  $i$  da rede, não necessariamente esse nó possui alcance suficiente para se comunicar com todos os outros nós, e é o que na realidade normalmente acontece. Um nó  $i$  é chamado nó escondido, se  $i$  não consegue se comunicar diretamente com um outro nó  $j$ , e as transmissões tanto de  $i$  como de  $j$  causam interferência entre si, se forem executadas ao mesmo tempo (RAY et al., 2005; JIANG e LIEW, 2007).

A figura 23 (a) ilustra um caso de nós escondidos. O nó A não transmite ao nó C. Nesse caso, o nó B está no raio de cobertura tanto do nó A quanto do nó C. Logo, quando A está transmitindo para B, qualquer transmissão de C irá interferir na transmissão entre A e B. Na figura 23 (a) é ilustrado o caso em que A está transmitindo para B quando C começa a transmitir para D, interferindo na transmissão do primeiro.

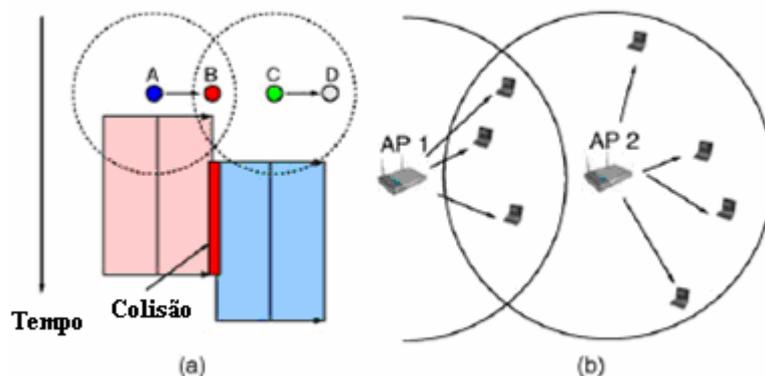


Figura 23: (a) As antenas A e C não estão na mesma área, mas conflitam quando transmitem; (b) Rede *wireless* com os servidores AP1 e AP2 escondidos, mas com clientes na mesma área (RAY et al., 2005).

Esse caso é conhecido na literatura de redes *wireless* como transmissão concorrente (*concurrent transmission*) (GANDHAM et al., 2008). Ele ocorre quando as transmissões são feitas no mesmo canal que fica, portanto, ocupado. A figura 23 (b) ilustra o caso em que mesmo os clientes estando ligados ao servidor AP1, eles estão na área de atuação de AP2 e portanto, AP2 pode interferir no funcionamento entre AP1 e os clientes.

Para administrar esse problema de forma a não impactar na qualidade do tráfego da rede, pode-se controlar a potência de algumas antenas (LI; WANG, 2008), aumentar a diversidade de antenas, utilizar *softwares* para protocolos de rede (GHABOOSI et al., 2008) ou mover os nós ou remover obstáculos. Ekici e Yongacoglu (2008) afirmam que em condições médias de tráfego, os nós escondidos podem diminuir em 33% o desempenho da rede.

Uma das estratégias utilizadas para gerenciar as transmissões concorrentes é a utilização de sensores que controlam os momentos em que as antenas podem ou não podem transmitir (YICK et al., 2008). De certa forma, esses sensores atuam como semáforos, controlando o tráfego de informações na rede. A consequência disso é que o problema de transmissão em redes com nós escondidos pode ser entendido como um problema de programação (*scheduling*) com restrições de adjacência. A rede de nós é um recurso, que possui diversos postos de trabalho, que são os nós que transmitem e recebem as informações, com uma duração determinada, ou seja, essas últimas podem ser entendidas como tarefas. Então, as redes de transmissão são multiprocessadores que processam as transmissões de informações ou, as tarefas.

As tarefas que são processadas possuem um conjunto de restrições que, quando estão em execução, bloqueiam a utilização de outros recursos, que são as antenas que sofrem transmissão concorrente, caracterizando assim, a restrição de adjacência. A figura 24 (a) apresenta o grafo que conecta os nós da rede apresentada na figura 24 (b).

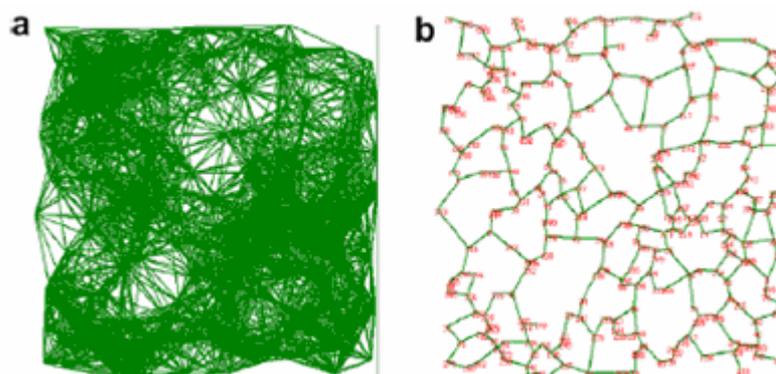


Figura 24: Exemplo de uma rede com 100 nós (SHEU et al., 2008).

As relações entre os nós podem ser representadas por um grafo de conflito, que restringe a transmissão simultânea entre alguns pares de nós. Diferentemente do planejamento de florestas, no problema de programação da transmissão de redes de comunicação com nós escondidos e transmissões concorrentes, os recursos que serão utilizados durante a execução das tarefas não são conhecidos previamente, mas apenas a origem e o destino da transmissão. A escolha dos diversos caminhos (ou conjunto de nós que formam o conjunto de estações de trabalho) através da rede depende da programação utilizada. Na figura 24 (b) fica claro ao se escolher quaisquer dois pontos para transmitir um pacote de informação, existem vários caminhos que podem ser seguidos além do caminho direto, que pode estar bloqueado.

O problema de programação da transmissão de redes, da mesma forma que o problema de planejamento da colheita de madeira em florestas, possui uma série de outras

restrições específicas ao problema que fogem do escopo deste trabalho. Segundo Ghaboosi et al. (2008), esse problema foi estudado primeiramente por Kleinrock e Tobagi (1975 a; b) no controle de transmissão de rádios portáteis. Kleinrock e Tobagi (1975 a; b) sugerem a utilização de um tom de ocupado para rádios que estão em uso ou bloqueados por transmissões.

Recentemente, esse problema voltou a ser estudado devido ao desenvolvimento de redes *wireless*, como já citado, e continua a ser um problema que desperta grande interesse (TSERTOU, LAURENSEN, 2008). As principais abordagens de solução para os problemas reais são a utilização de algoritmos (GHABOOSI et al., 2008) e simulação estocástica para estudar e programar as transmissões. A simulação é utilizada nos casos em que os problemas reais possuem variações aleatórias de volume, taxa de chegada e processamento de tráfego, que podem ser tratados com a utilização da teoria de filas (RAY et al., 2004). Um outro exemplo de problema de nós escondidos aplicados pode ser encontrado em Jiang e Liew (2007), que estudaram como minimizar os efeitos dos nós escondidos em redes de telefonia celular.

Contudo, foram Even e Shanar (2006) os primeiros, aparentemente, que propuseram que os problemas de programação de pacotes em redes de transmissão de antenas fossem vistos como um problema de programação com conflitos (*scheduling with conflicts*), ao menos para a revisão bibliográfica realizada nesse estudo. Depois, Kaplan (2007) e Even et al. (2009) continuaram estudando os problemas de programação com conflitos, que para este estudo, são os problemas de programação com restrição de adjacência. Even e Shanar (2006) referem-se a outros problemas de programação com restrição de adjacência, como em Jansen (1999; 2003) e Epstein e Levin (2007; 2008) que são problemas de *bin packing* com restrições especiais.

Irani e Leung (1996; 2003) estudam o problema de programar cruzamentos controlados por semáforos, como o representado na figura 25 que também pode ser entendido como um problema de programação com restrições de adjacência. Considere que cada grupo de carros que deve passar pelo semáforo é uma tarefa, e cada uma das direções que o grupo de carros segue é um recurso como uma máquina. Assim, por exemplo, um grupo de carros que deve atravessar o cruzamento na direção e sentido 2 no cruzamento da figura 25 pode ser entendido como uma tarefa que deve ser processada pela estação de trabalho 2.

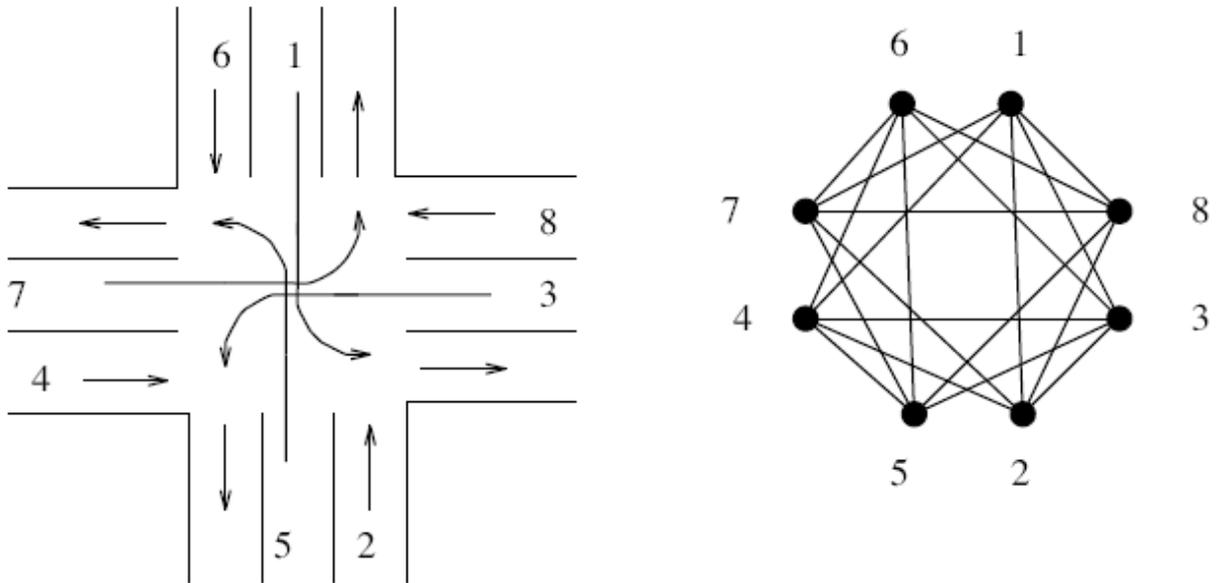


Figura 25: Representação de um cruzamento comandado por um semáforo e o respectivo grafo de conflito (IRANI; LEUNG, 2003).

Como qualquer cruzamento comum, quando um sentido está aberto ao tráfego, os outros sentidos perpendiculares devem estar fechados para permitir o tráfego. As diversas direções que não são possíveis de programar ao mesmo tempo são representadas pelo grafo de conflito da figura 25. Nesse caso, o cruzamento pode ser entendido como um multiprocessador, com diversas estações de trabalho em que, tarefas que estão sendo processadas em uma dada estação de trabalho interferem em outras tarefas que são processadas em outras estações de trabalho, caracterizando assim a restrição de adjacência. Nesse caso, Irani e Leung (1996, 2003) chamam o problema de programação com conflitos entre tarefas e propõem métodos heurísticos para a solução do problema.

Duin e Sluis (2006) apresentam algumas aplicações dos modelos de programação de projetos a problemas de programação com restrições de adjacência. Considere que existe um balcão de atendimento de uma companhia aérea em um aeroporto com diversos postos de trabalho que podem ser alocados funcionários. Normalmente, as companhias aéreas mantêm alguns postos dedicados ao atendimento em geral e, conforme a hora de partida, a quantidade esperada de passageiros e a natureza do voo, a empresa aloca um conjunto de estações do balcão de atendimento para atender apenas um voo durante períodos do dia. A alocação de postos de trabalho é feita sempre em postos adjacentes, ou seja, postos vizinhos. Considerando que cada voo é uma tarefa, existe um problema em alocar a quantidade de postos de trabalho adjacentes ao longo de um dia inteiro, considerando que os

postos devem ser sempre adjacentes e atender à demanda de tarefas a serem cumpridas com prazos de entrega.

Um exemplo de programação de recursos adjacentes é apresentado na figura 26. Considere que o eixo horizontal da figura representa as unidades de tempo e no eixo vertical representa os postos de trabalho dos atendentes das companhias aéreas. A figura 26 (a) caracteriza uma configuração de tarefas infactível, já que a tarefa 6 não está distribuída em postos adjacentes. Na figura 26 (b) é apresentado um programa alternativo de atendimento viável.

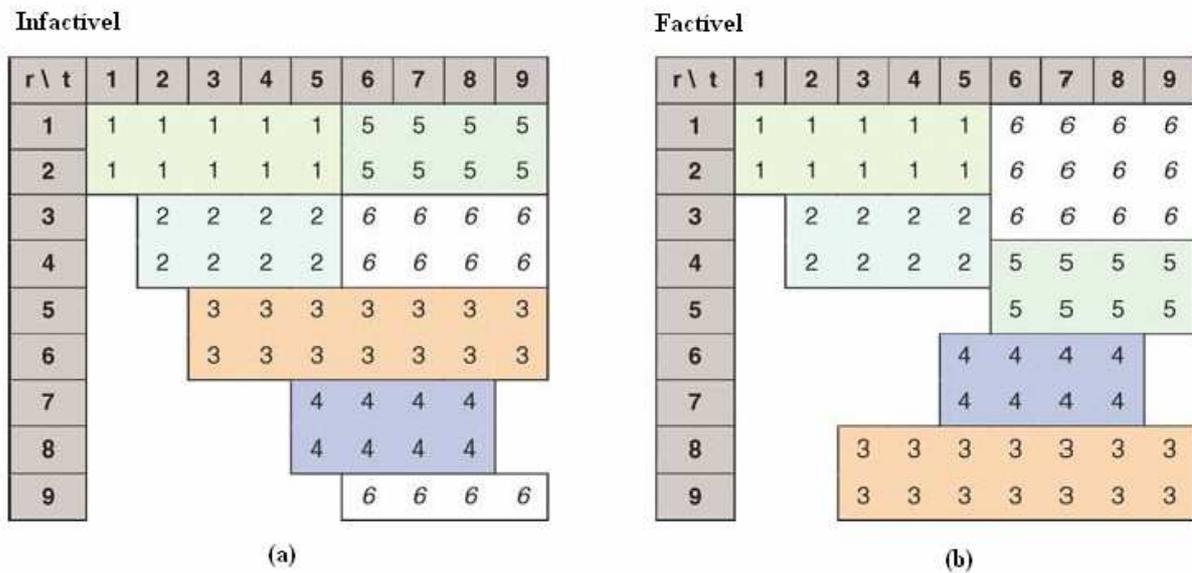


Figura 26: Exemplo de programação de recursos adjacentes (DUIN; SLUIS, 2006).

Schäffer (1997) estuda o mesmo tipo de problema, chamado por ele de *forbidden sets*. Considere um conjunto de  $n$  tarefas que necessitam do mesmo tipo de máquinas e que existem  $k$  máquinas deste tipo. Então, no máximo  $k$  máquinas dessas  $n$  tarefas poderiam ser programadas ao mesmo tempo (SCHÄFFER, 1997). Schäffer (1997) está estudando o problema de programar multiprocessadores com restrições de adjacência, com todas as estações de trabalho sendo idênticas e tarefas que utilizam mais de uma estação de trabalho ao mesmo tempo.

A semelhança entre problemas de programação de projetos com restrições de recursos e os problemas de programação de multiprocessadores foram identificados por diversos trabalhos como Brucker e Krämer (1996) e Drozdowski (1995, 1996). Drozdowski (1996) afirma que os trabalhos com multiprocessadores ainda fomentam diversas pesquisas para poderem ser entendidos. Drozdowski (1995) comenta que os problemas clássicos de programação da produção consideram que as tarefas são executadas em apenas um recurso de

cada vez. Os problemas de programação de projetos com restrições de recursos são uma categoria de problemas que consideram a utilização de vários recursos ao mesmo tempo por uma mesma tarefa.

Os estudos de programação de projetos com restrições de recursos encontrados na pesquisa realizada por este trabalho não encontrou casos em que se estuda a influência das relações entre as tarefas, ou como as tarefas que utilizam ou bloqueiam recursos em comum se interferem. Apesar da similaridade, os problemas de multiprocessadores e de programação de projetos com restrições de recursos encontrados não explicitam a restrição de adjacência entre as tarefas, por exemplo, na forma de grafo de conflito, como nas aplicações de telecomunicações ou de planejamento de colheita em florestas e rotação de culturas na agricultura.

Os problemas de corte e empacotamento (*cutting and packing problems*) surgem em várias aplicações, inclusive aplicações industriais. Os problemas de corte são os problemas que estudam como a partir de unidades grandes, pode-se dividi-las em unidades pequenas atendendo um conjunto de restrições e objetivos. Os problemas de empacotamento estudam como agrupar um conjunto de unidades pequenas em uma ou várias unidades grandes, sujeito a uma série de restrições e objetivos (DYCKHOFF, 1990).

Entre os problemas de corte e empacotamento estão os problemas de empacotamento de *bin* com formatos e tamanhos diferentes, a um conjunto de *bins* idênticos, de modo a minimizar a quantidade de *bins* utilizados (WÄCHER et al., 2007). Uma das formas generalizadas do problema de *bin packing* é o *bin packing* com conflitos (*bin packing with conflicts*). Jansen (1999) define os problemas de *bin packing* com conflitos, no caso em que os recursos (ou máquinas) são idênticos, como dado um grafo não dirigido  $G = (V, E)$  com o conjunto de itens  $V = \{1, \dots, n\}$  e tamanhos  $s_1, \dots, s_n$  que são números racionais no intervalo  $(0, 1]$ , o problema é particionar o conjunto  $V$  de itens em *bins* independentes  $U_1, \dots, U_h$  com  $\sum_{i \in U_j} s_i \leq 1$  para cada  $1 \leq j \leq h$ , respeitando as relações de conflito do grafo  $G = (V, E)$ , utilizando um número mínimo de *bins*.

Jansen (1999) ainda afirma que o problema de programação de tarefas com exclusão mútua de Baker e Coffamn (1996) é um caso especial do *bin packing* com conflitos. Jansen (1999) define que:

Se  $J$  é o conjunto de tarefas restritas pela mútua exclusão pelo grafo  $G$  em que cada vértice representa uma tarefa que requer uma unidade de tempo. As tarefas devem ser programadas sem preempção em  $m$  processadores idênticos com as tarefas

adjacentes em  $G$  ocorrendo em intervalos disjuntos. Ao se representar a unidade de tempo como um *bin*, então a capacidade de cada bin é  $m$  (o número de máquinas) e os itens devem ser empacotados em um número mínimo de *bins* (*makespan*) sem conflitos.

Assim, para este trabalho os problemas de programação de tarefas com mútua exclusão são problemas de programação de tarefas com restrições de adjacência, então, os problemas de *bin packing* com conflitos também são problemas de programação de tarefas com restrições de adjacência, representados pelo grafo de conflito definido por Jansen (1999). Jansen (1999) ainda afirma que os problemas de *bin packing* com conflitos são aplicáveis à problemas de programação, telecomunicações, armazenamento de arquivos ou a programação de horários em escolas. Os problemas de *bin packing* com conflitos são uma generalização dos problemas de *bin packing* e são objetos de estudo em várias aplicações (por exemplo, Gendreau et al. (2004), Epstein e Levin (2007), Epstein et al. (2008)).

### 3.6 Curvas de Aprendizagem

A curva de aprendizagem é um fenômeno empírico observado pela primeira vez na indústria aeronáutica, na produção de aviões. Nahmias (2008) afirma que foi na década de 1920 que a indústria aeronáutica descobriu que a quantidade de mão de obra necessária para produzir uma unidade diminui a cada unidade produzida. Nahmias (2008) afirma ainda que esse efeito foi observado em outras indústrias e que a diminuição marginal da quantidade de mão de obra por unidade acumulada era decrescente.

Krajewski e Ritzman (1996) e Schemenner (1990) afirmam que as curvas de aprendizagem foram desenvolvidas apenas durante a Segunda Guerra Mundial, também na indústria aeronáutica, principalmente na produção de estruturas aeronáuticas. Nahmias (2008) comenta que tipicamente, a relação entre a quantidade de mão de obra em homem-hora  $Y$  necessária para produzir a unidade número  $u$  é exponencial e pode ser representada pela seguinte expressão:

$$Y(u) = a.u^{-b} \quad (39)$$

em que  $a$  é a quantidade de horas-homem necessária para produzir a primeira unidade e  $b$  é um parâmetro empírico relacionado a taxa com que  $Y(u)$  decai a cada unidade produzida. Comumente, as curvas de aprendizagem são classificadas conforme a quantidade que  $Y(u)$  decresce cada vez que o número de unidades produzidas,  $n$ , dobra. Assim, explica Nahmias

(2008), uma curva de aprendizagem de 80% significa que  $Y(u)$  diminui 80% a cada  $2n$  unidades produzidas. Na expressão (40) é apresentado como esse resultado pode ser obtido a partir da expressão (39).

$$\frac{Y(2u) = a \cdot 2u^{-b}}{Y(u) = a \cdot u^{-b}} = 2^{-b} = 0,8 \quad (40)$$

Deste modo,  $b$  pode ser calculado pela expressão (41).

$$b = -\frac{\ln(L)}{\ln(2)} \quad (41)$$

em que  $L$  é a porcentagem da curva de aprendizado, também chamada de taxa de aprendizagem, com valores entre 0 e 1. A figura 27 ilustra um exemplo de curva de aprendizagem de 80% com custo de 250 homens-hora para produzir a primeira unidade.

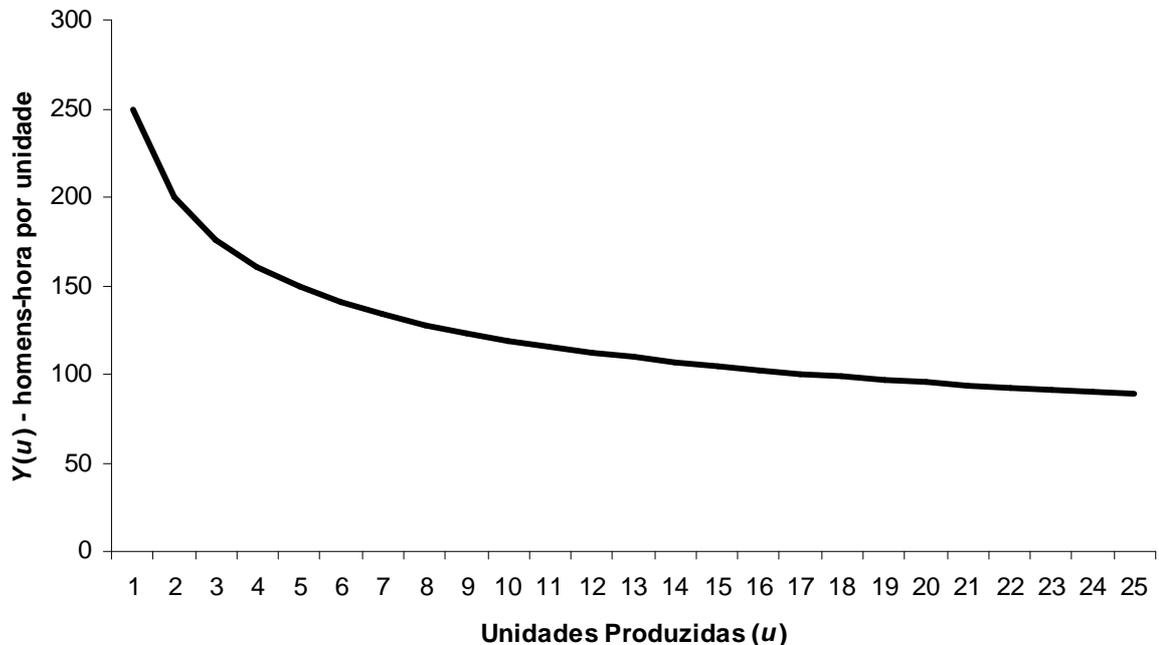


Figura 27: Curva de aprendizagem de 80%

A primeira e talvez mais intuitiva explicação para a diminuição da quantidade de homem-hora com o aumento da produção acumulada é o ganho de experiência dos operadores. No entanto, Schmenner (1990) afirma que diversos fatores relacionados a três aspectos da produção definem a curva de aprendizagem. O primeiro aspecto é mão de obra. Além do próprio aprendizado associado à produção repetida de diversas unidades, novas formas de organização do trabalho e até pressão dos administradores podem fazer o custo por unidade diminuir. O segundo aspecto são modificações no processo de produção como melhorias nos métodos de produção, automação, substituição de equipamentos, integração

vertical e avanços tecnológicos. O terceiro aspecto são as modificações no produto. Substituição por materiais mais baratos ou com processos de fabricação mais simples, além de alterações no projeto, permite reduzir custos de produção também.

Nahmias (2008) também comenta que a melhoria em processos de programação e controle de estoques e organização da área de trabalho também contribui para a diminuição do custo de produção. Assim, a curva de aprendizagem é um fenômeno mais acentuado em produtos mais complexos, que oferecem mais oportunidades de modificações dos três aspectos mencionados. Também, é mais útil nas fases de projeto e início de produção (KRAJEWSKI; RITZMAN, 1996). Nahmias (2008) afirma que a diminuição de custos pode não ser ilimitada e que, dependendo do produto e processo produtivo, pode chegar a um limite em que a redução de custo com a produção é nula.

Krajewski e Ritzman (1996) enumeram diversas aplicações da curva de aprendizagem. Elas podem ser utilizadas no planejamento de recursos de projetos, permitindo estimar a quantidade de mão de obra que será necessária ao longo do tempo, na variação do custo unitário ao longo do tempo, no planejamento financeiro, ajudando na determinação dos fluxos de caixa da operação no futuro, e também no planejamento de mão de obra, considerando as contratações e treinamentos que serão necessários. Schmenner (1990) aborda as curvas de aprendizagem em um contexto mais amplo, considerando a evolução da aprendizagem como parte de um processo importante de gestão da inovação, tanto de produtos como de processos de produção.

Nahmias (2008) alerta que Henry Ford tinha como estratégia inicial para diminuir o custo de seu modelo T entre 1909 e 1923 justamente produzir em alto volume para, entre outros motivos, percorrer a curva de aprendizagem mais rapidamente e assim diminuir seu custo unitário. No entanto, essa estratégia se mostrou limitada, pois, no longo prazo, perde-se a capacidade de se adaptar a mudanças no mercado, caso a ênfase em padronização seja muito rígida. Nahmias (2008) e Krajewski e Ritzman (1996) citam também um exemplo bastante conhecido de aplicação de curvas de aprendizagem na indústria aeronáutica, que é o caso da Douglas Aircraft. Durante a década de 1960, curvas de aprendizagem na produção de estruturas aeronáuticas, que acabaram por se mostrar muito otimistas em relação à realidade de produção, causaram sérios problemas financeiros à empresa. Isso levou a Douglas Aircraft a ser comprada pela McDonnell Company, formando a McDonnell-Douglas, que mais tarde foi comprada pela Boeing.

Anzanello e Flogliatto (2007) apresentam uma ampla revisão sobre curvas de aprendizado, incluindo representações matemáticas diferentes das apresentadas nesta seção, e

os efeitos que a interrupção ou introdução de inovações na produção de um produto tem sobre a curva de aprendizagem. Porém, essas abordagens fogem do escopo deste trabalho e por isso não são apresentadas.

## 4 Descrição do Problema e Modelagem Matemática

### 4.1 Apresentação do problema

Esse trabalho estuda o problema de programação da montagem de subconjuntos de aviões jato, presente em uma indústria aeronáutica. É estudado um subconjunto de uma aeronave presente em uma indústria aeronáutica que utiliza um gabarito com restrições de adjacência.

#### 4.1.1 Montagem do subconjunto

O subconjunto é formado por oito partes, chamadas de peças, que são montadas individualmente em um gabarito múltiplo, que é capaz de montar todas as peças do subconjunto em oito postos de trabalho adjacentes e específicos de cada peça do subconjunto. A figura 28 apresenta o subconjunto e as suas respectivas peças. Cada parte é chamada por um número, entre 1 e 8, que identifica a peça. A programação nesse caso é feita de modo manual sob responsabilidade da própria produção. O setor responsável pelo planejamento e controle da produção da empresa programa a produção apenas em nível mais agregado, estabelecendo cadências de produção dos aviões e as datas de entrega dos subconjuntos referentes a cada avião.

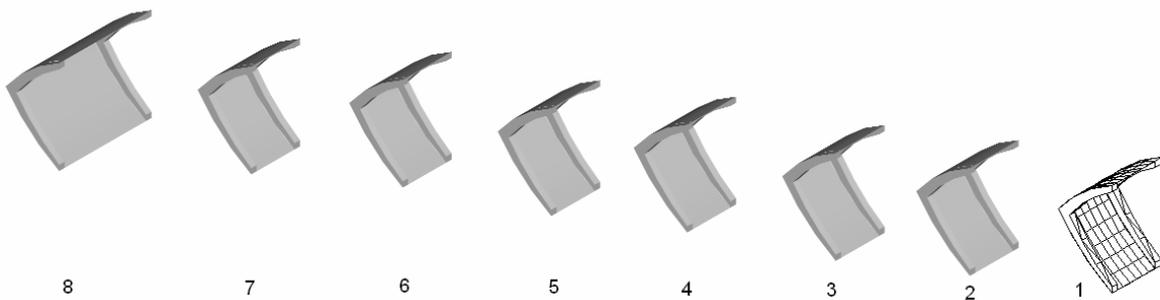


Figura 28: Subconjunto e as suas respectivas peças

A figura 29 apresenta o gabarito e as 8 estações de trabalho que são utilizadas para montar as peças do subconjunto. Cada estação de trabalho recebe um nome, entre 1 e 8, que representa exatamente a peça que deve ser montada naquela estação. Assim, a estação 1 monta a peça 1, a estação 2 monta a peça 2, e assim por diante até a estação 8.

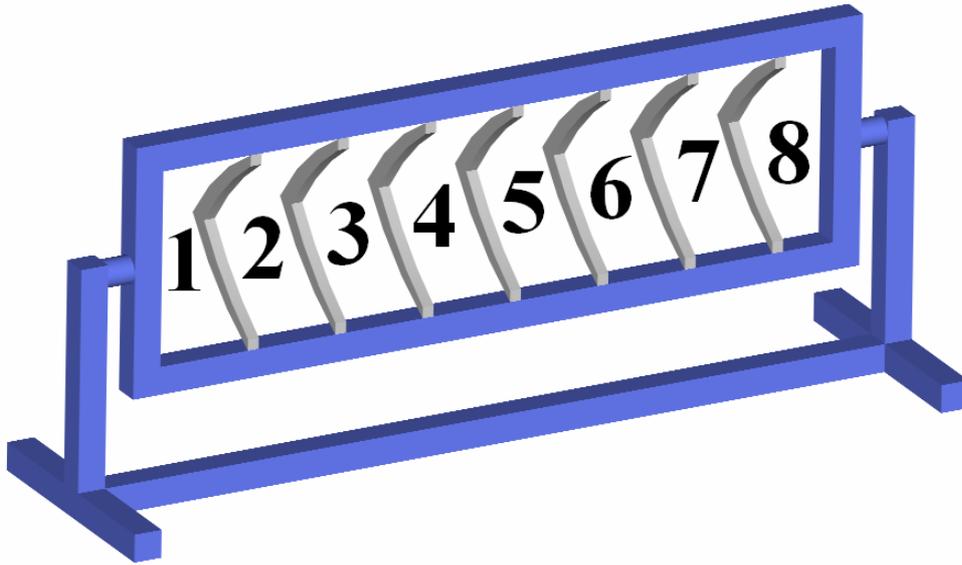


Figura 29: Gabarito múltiplo com restrições de adjacência utilizado na montagem do subconjunto

A montagem das peças no subconjunto é totalmente manual e feita em duas etapas, com duas operações cada. Cada etapa é uma tarefa, formada por uma operação de montagem: uma realizada no gabarito e outra realizada em bancadas para trabalhos manuais, próximas ao gabarito. A figura 30 representa a rede de tarefas (ou atividades) e suas relações de precedência, com as atividades apresentadas nos nós. Cada tarefa então, recebe um nome numérico e um outro dígito. Esse dígito, que pode ser 1 ou 2, é referente à operação da tarefa. Assim, dígito 1 significa operação de gabarito e dígito 2 significa operação de bancada. Assim, nesse caso, um subconjunto com 8 peças precisa de 16 tarefas e 32 operações para ser montado por completo.

Como cada avião precisa de 2 subconjuntos desses, um avião precisa de 32 tarefas e 64 operações para ficar pronto. O número de operações varia conforme a empresa percorre a curva de aprendizagem e evolui o processo de produção, podendo aumentar ou diminuir, visando sempre alguma melhoria no processo de produção.

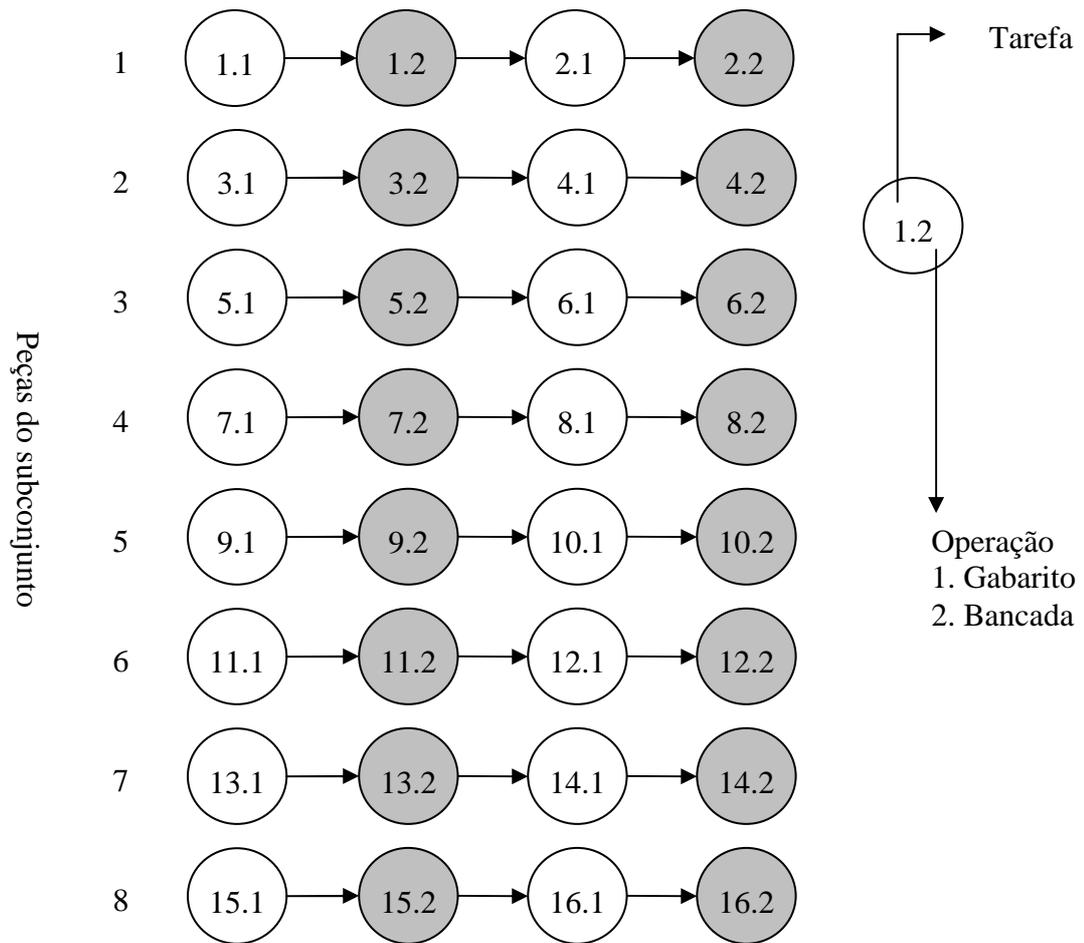


Figura 30: Rede de relações das tarefas de montagem do subconjunto

Por limitações de espaço físico, quando um operador está utilizando uma estação de trabalho, por exemplo, a estação 4, as estações 3 e 5 não podem ser utilizadas. Assim, caracteriza-se a restrição de adjacência entre as tarefas das peças 3, 4 e 5. A figura 31 apresenta o grafo de adjacência, em que cada estação de trabalho corresponde a um nó e os arcos ligam os nós que possuem restrição de adjacência entre si.

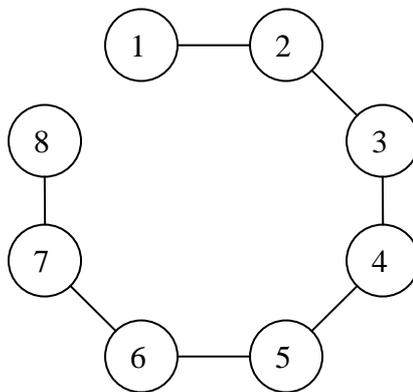


Figura 31: Grafo de adjacência das estações de trabalho

A figura 32 apresenta o grafo da relação de adjacência das 16 tarefas apresentadas na figura 30 que formam um subconjunto de um avião. Para 2 aviões, basta adicionar mais 16 tarefas com as mesmas relações de adjacência entre si, além das relações de adjacência entre dois revestimentos diferentes.

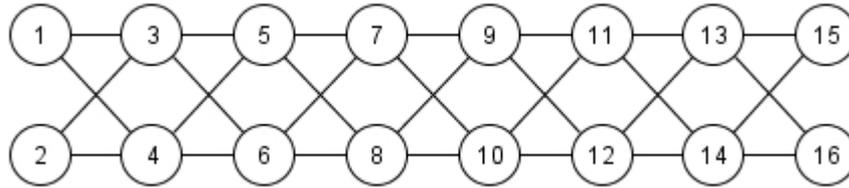


Figura 32: Grafo das relações de adjacência das tarefas de dois subconjuntos inteiros

Um outro subconjunto é formado por 7 peças, que são montadas em um gabarito múltiplo com 7 postos de trabalho adjacentes. A figura 33 apresenta as peças que formam esse outro subconjunto, numeradas de 1 a 7. Assim como no caso anterior, a peça 1 é montada no posto de trabalho 1, a peça 2 é montada no posto de trabalho 2, e assim por diante, conforme o gabarito apresentado na figura 34.

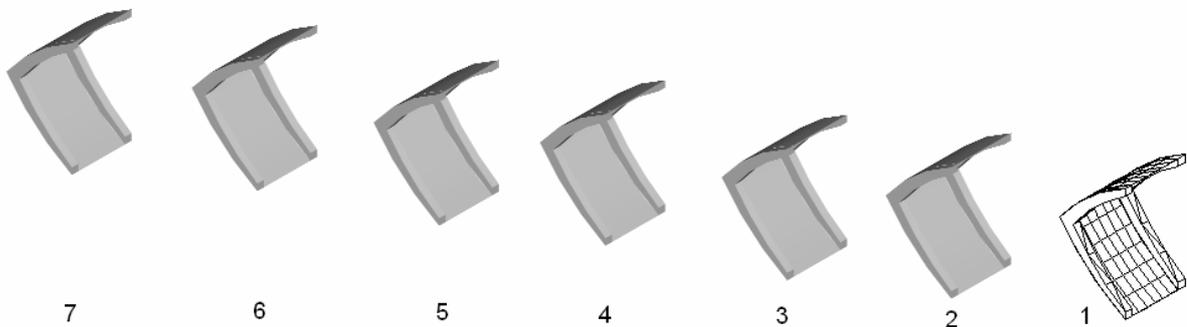


Figura 33: Segundo subconjunto e as suas respectivas peças

Conforme mencionado acima, o gabarito onde o segundo subconjunto é montado possui 7 postos de trabalho adjacentes. Observe que, nesse caso, os postos de trabalho 1 a 4 estão no lado oposto no gabarito dos postos de trabalho 5 a 7. O espaço entre os postos de trabalho é menor que no gabarito do primeiro subconjunto. Logo, cada montagem que é realizada em qualquer posto de um dos lados bloqueia todas as estações daquele lado. Por exemplo, qualquer peça que é montada em um dos postos de trabalho 1 a 4 bloqueia os outros postos de trabalho deste lado, mas não bloqueia os postos de trabalho 5 a 7. Assim, nesse caso, as restrições de adjacência são tão fortes que o gabarito pode ser abordado como se tivesse apenas dois postos de trabalho, formados pelos postos 1 a 4 e 5 a 7 respectivamente.

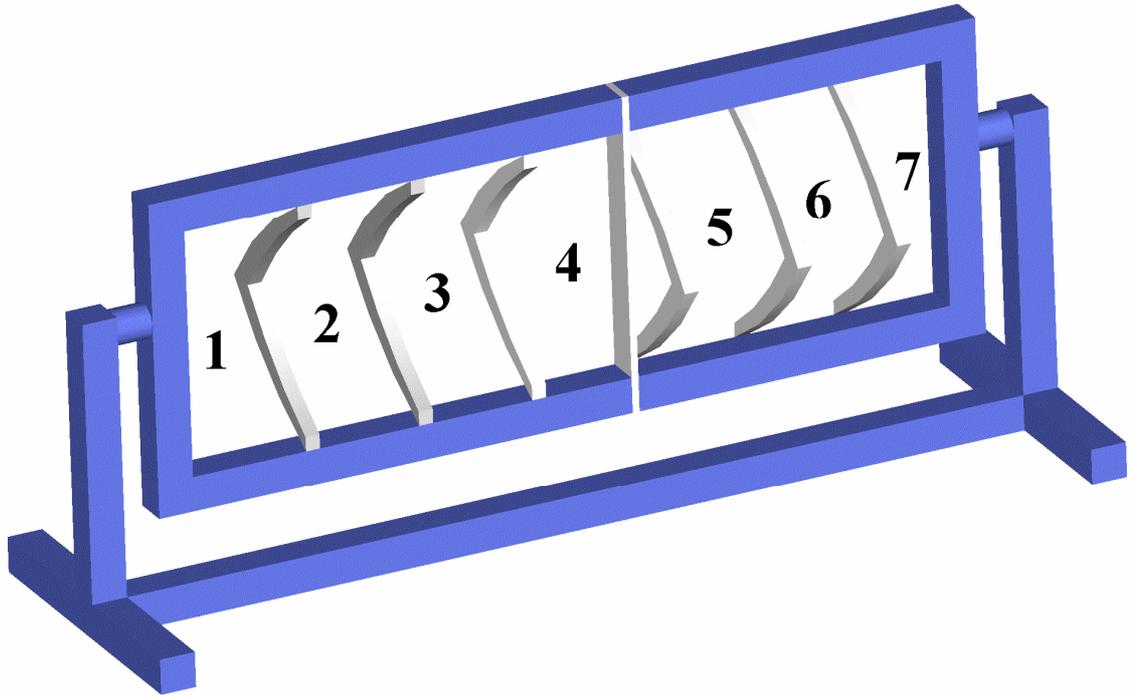


Figura 34: Gabarito múltiplo com restrições de adjacência utilizado na montagem do subconjunto 2

As montagens do subconjunto 2 também são manuais e tem duas etapas, uma de gabarito e outra de bancada. Entretanto, a sua rede de precedência de tarefas é diferente da rede do primeiro subconjunto, apresentada na figura 30. A figura 35 apresenta a rede de precedência da montagem do subconjunto 2. As duas primeiras operações de montagem das peças podem ser realizadas em paralelo. Conforme anteriormente, as operações em cinza são realizadas nas bancadas e as operações em branco são realizadas no gabarito. Além disso, todas as peças devem ser unidas numa etapa final de montagem do subconjunto. Como cada avião precisa de dois desses subconjuntos e cada subconjunto é formado por 15 tarefas, logo, um avião precisa de 30 tarefas para ficar pronto.

A figura 36 apresenta o grafo de conflito entre os postos de trabalho do gabarito. Como alguns postos de trabalho estão em lados opostos do gabarito, o grafo é composto de dois subgrafos que não possuem nós ligados por arcos (i. e., são desconexos). A figura 37 apresenta o grafo de conflito entre as tarefas que formam um subconjunto. Nesse caso, também devido às estações estarem em lados opostos do gabarito, o grafo é composto de dois subgrafos desconexos em que os nós não são conectados por nenhum arco entre si.

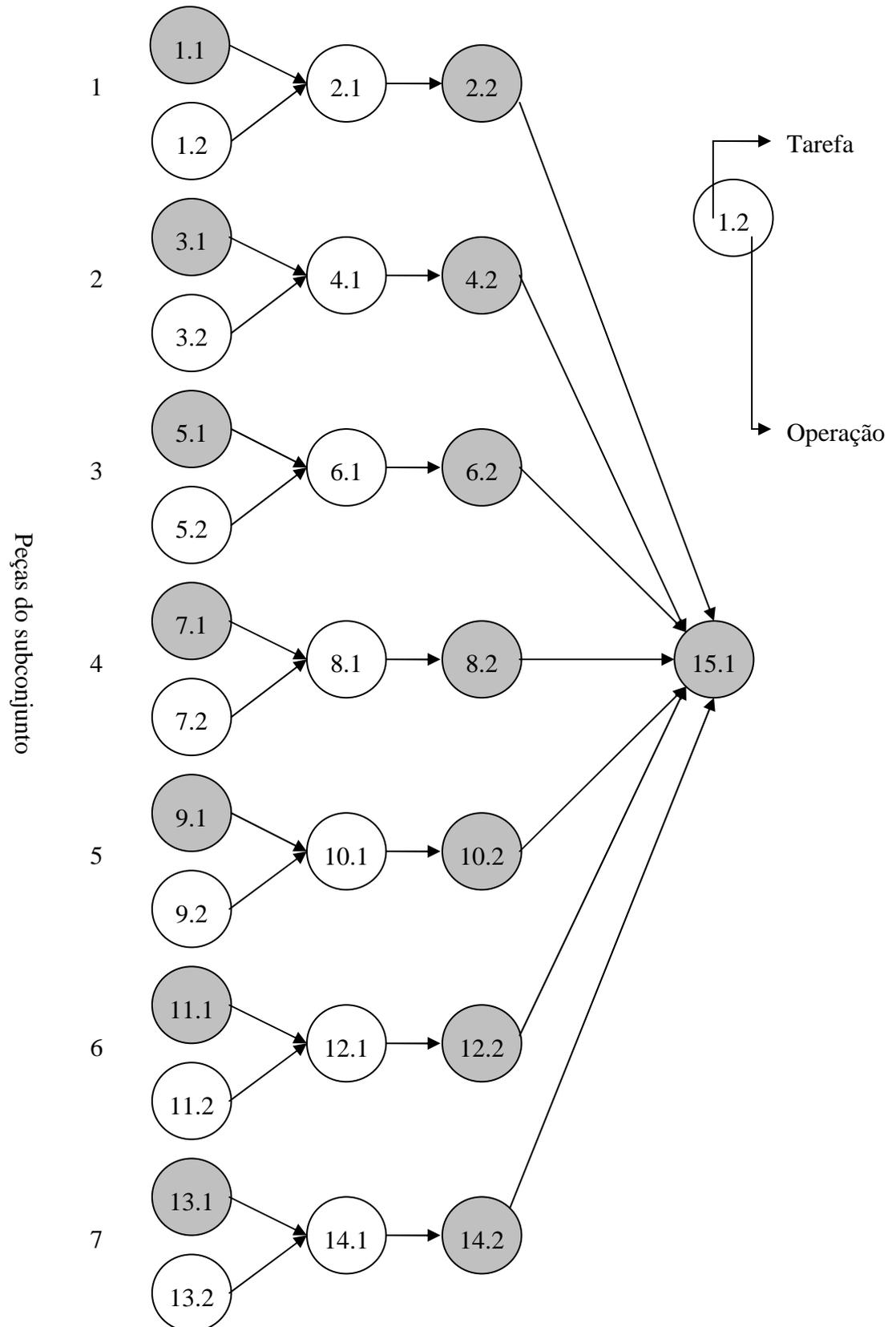


Figura 35: Rede de relações das tarefas de montagem

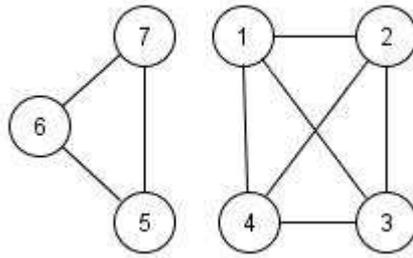


Figura 36: Grafo de adjacência das estações de trabalho

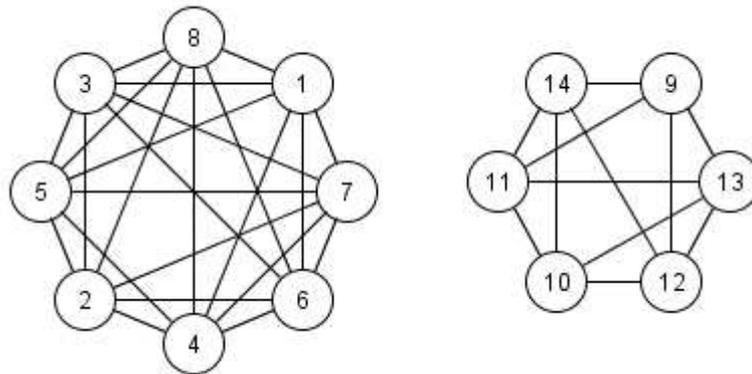


Figura 37: Grafo das relações de adjacência das tarefas do segundo subconjunto inteiro

O segundo subconjunto, embora tenha o processo de montagem similar ao primeiro subconjunto, possui relações de adjacência tão fortes, que é possível abordar cada um dos dois lados do gabarito como estações únicas e independentes, ou seja, pode-se entender esse gabarito como dois gabaritos flexíveis. Por isso, a programação da montagem no segundo gabarito não foi estudada nesse trabalho.

#### 4.1.2 Estratégias de produção por avião e por peça

Para produzir as peças, duas estratégias podem ser adotadas. A primeira seria tratar as peças de cada subconjunto de forma individual e assim, trabalhar com um número maior de peças em produção ao mesmo tempo. Cada avião possui dois subconjuntos com 8 peças cada. Na prática isso significa que é permitido produzir vários aviões diferentes ao mesmo tempo. Essa é a programação da montagem de subconjuntos por peça.

A segunda estratégia seria programar a montagem dos subconjuntos de um avião de cada vez. Isso significa que, para uma dada cadência de produção, se distribui as peças relativas a cada avião no mesmo período e se produz um avião de cada vez, até completar o tempo de ciclo. Essa é a programação da montagem de subconjuntos por avião.

A cadência de produção é a quantidade de aviões de um modelo que é produzida por mês, quantidade esta que pode variar de algumas unidades a algumas dezenas de aviões por mês. O tempo de ciclo é o intervalo de tempo entre o término ou início da produção de dois aviões sucessivos e é relacionado à cadência de produção,  $C$ , e o tempo disponível,  $T$ , da seguinte forma:

$$t_{ciclo} = \frac{T}{C}$$

Assim, se a cadência de produção é de 10 aviões por mês e a empresa tem 20 dias úteis disponíveis por mês, o tempo de ciclo deve ser:

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

Isso significa que, caso um observador se posicione ao final da linha de produção de aviões, deverá contar um avião pronto a cada 2 dias para atender ao plano de produção sem formação de estoques desnecessários. A figura 38 ilustra as duas estratégias. Os números no eixo vertical representam o número de série do avião. Assim, o avião começa a ser produzido quando a primeira operação de uma das suas peças é iniciada e termina somente quando a última operação da última peça é realizada. Por isso, na primeira estratégia os tempos de montagem de cada avião variam e podem durar significativamente mais do que na segunda estratégia, cujos tempos de montagem de cada avião são previamente conhecidos segundo um programa de produção e o tempo de ciclo definido.

A vantagem da primeira estratégia em relação à segunda é que ela permite uma utilização melhor dos recursos por ter uma quantidade de peças maior em produção. No entanto, as desvantagens são maior estoque em processo, com todos os custos associados, como maior necessidade de pessoal de apoio para administrar a chegada e saída de peças da produção, maior frequência de perdas ou danos às peças componentes das peças em montagem e, conseqüentemente, e maior quantidade de retrabalho.

Por todas essas desvantagens, que na verdade são as vantagens da segunda estratégia, a estratégia de montagem por avião é adotada nesse trabalho. Além do que já foi citado, a estratégia por avião facilita o fluxo de produção, diminui o *leadtime*, permite otimizar o *layout* produtivo e acompanhar a situação das montagens mais fácil e rápido. A empresa onde a montagem estudada ocorre utiliza, nessa montagem em específico, a estratégia por peça por não ter um programa de produção definido.

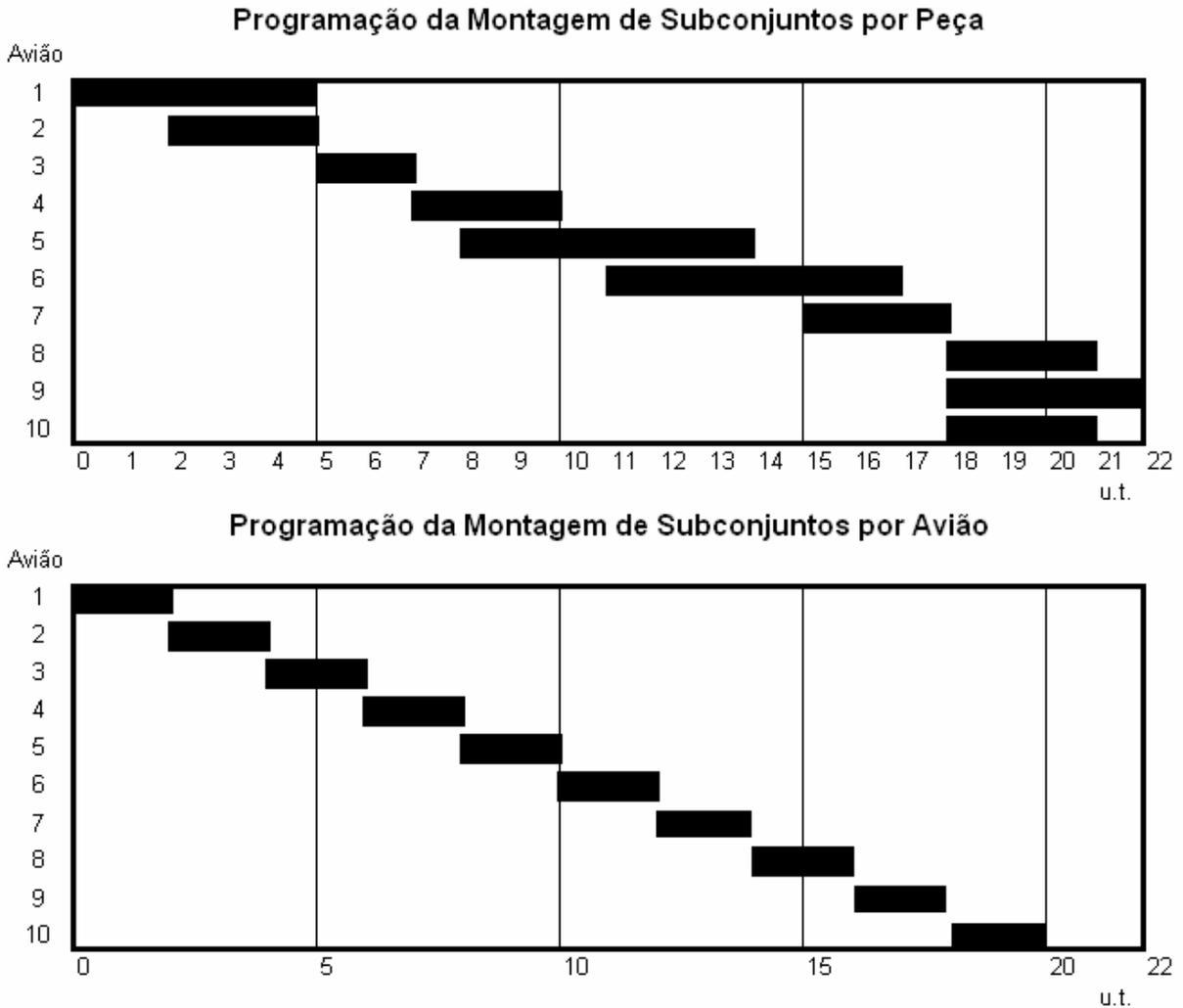


Figura 38: As duas estratégias de programação da produção dos aviões

Como o processo de produção de estruturas aeronáuticas muda conforme a curva de aprendizagem é percorrida, a montagem do subconjunto apresentado é considerada em quatro situações específicas, ou quatro fases. A curva de aprendizagem considerada nesse trabalho enfoca principalmente nas mudanças que acontecem na equipe de montagem, provenientes do treinamento da mão de obra. Não foi focado o formato da curva de aprendizagem ou a redução conforme o número acumulado de unidades produzidas aumenta.

- **Fase 1:**

A fase 1 é a de projeto, quando os tempos de cada operação são simulados ou estimados baseados na produção de protótipos. Nessa fase, a programação está mais preocupada em descobrir qual a cadência máxima de produção de aviões que um único gabarito conseguiria suportar e qual a quantidade de mão de obra necessária para cumprir essa tarefa. Portanto, seria um modelo de programação com mão de obra ilimitada. Nesse caso, deve-se otimizar a duração total da montagem, ou seja, o *makespan*.

- **Fase 2:**

A fase 2 inicia a produção em série, com cadência e prazos de entrega definidos. Nesse caso, a quantidade de mão de obra disponível é limitada e o tempo máximo para completar as tarefas é conhecido e dado pelo tempo de ciclo, calculado baseado na cadência de produção. Na fase 2, as montagens que são realizadas no gabarito são mais técnicas e, por isso, precisam de operadores experientes. Geralmente, esses operadores são aqueles que participaram da produção dos aviões protótipos e estiveram envolvidos diretamente com os detalhes técnicos do processo de montagem, conhecendo todas as montagens no gabarito e nas bancadas. Na prática, os operadores experientes são os que realizam as montagens nos gabaritos, o que não ocupa todo o seu tempo de trabalho, e passam o tempo restante ensinando o processo de montagem nas bancadas aos operadores novos na montagem. Então, nesse caso, a mão de obra é limitada e dividida em dois grupos: a mão de obra especializada, capaz de montar apenas nos gabaritos, e a mão de obra não especializada, capaz de montar apenas nas bancadas. Quando a mão de obra especializada está ensinando a mão de obra não especializada, é considerado que apenas a mão de obra não especializada está executando a operação.

- **Fase 3:**

Na fase 3, a mão de obra especializada não precisa mais ensinar as montagens na bancada para a mão de obra não especializada, podendo, assim, trabalhar tanto nas montagens de gabarito quanto nas montagens de bancada. Porém, a mão de obra não especializada monta apenas nas bancadas e está aprendendo a montar nos gabaritos.

- **Fase 4:**

Na fase 4, todos os operadores conseguem realizar as montagens de gabarito e bancada e assim, não há mais a divisão em dois grupos. Esse caso seria o final da curva de aprendizagem, quando apenas pequenas melhorias no processo de montagem permitiriam simplificar e diminuir muito pouco os tempos das operações.

A figura 39 ilustra as fases apresentadas anteriormente e a curva de aprendizagem da montagem do subconjunto em questão.

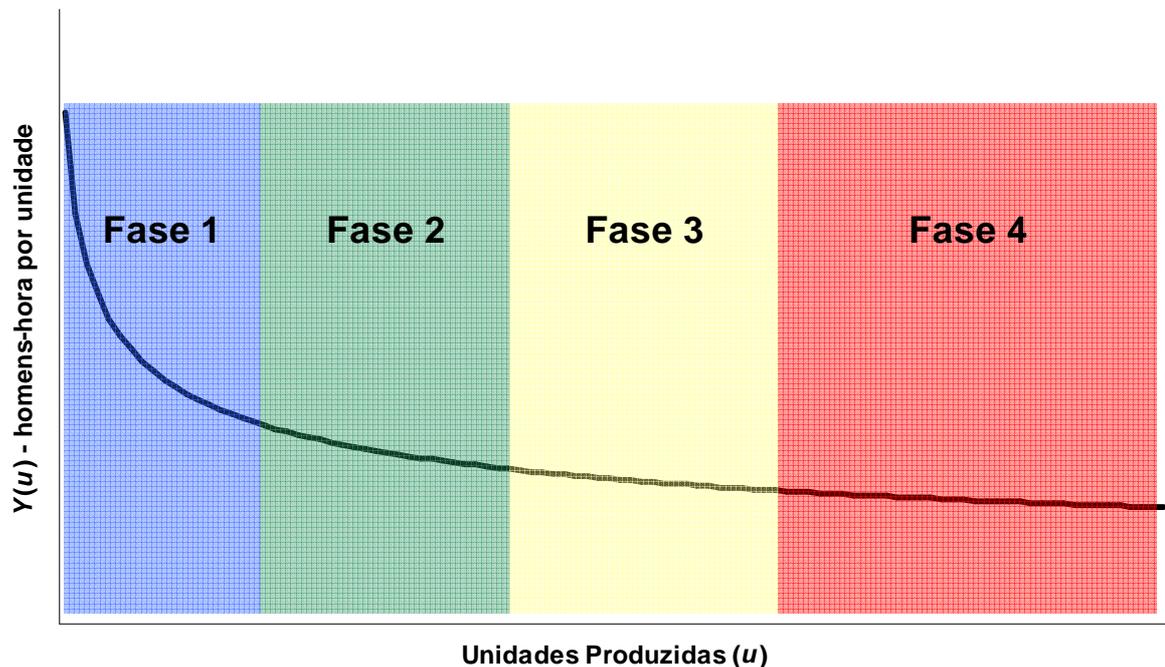


Figura 39: Fases da montagem enquanto percorre a curva de aprendizagem

No presente estudo, é desenvolvido um modelo matemático específico para cada uma das fases.

#### 4.2 Modelo matemático da fase 1

Como apresentado na seção 4.1, a fase 1 está preocupada com a programação do gabarito em fase de projeto e produção de protótipos, sem preocupação com a quantidade de mão de obra. A quantidade de mão de obra na solução ótima seria a máxima possível para a quantidade considerada de gabaritos. A mão de obra não é considerada uma limitação porque nessa fase pode-se contratar ou realocar pessoas de outras áreas, caso seja necessário. A maior preocupação nessa fase é saber quantos gabaritos serão necessários para atender às cadências de produção planejadas, já que os gabaritos demoram um tempo considerável para serem produzidos e tem um custo bastante significativo. A função objetivo é minimizar o tempo necessário para entregar todas as tarefas, ou seja, minimizar o *makespan*.

O modelo a seguir é baseado no modelo clássico de *job shop* apresentado em Baker (1974), Coffman (1967) e Morton e Pentico (1993), e descrito na seção 3.3, com mesma função objetivo. Então, define-se:

- $p_j$  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 às restrições de disponibilidade e prazos de entrega das tarefas, restrições de sequenciamento das tarefas em cada posto de trabalho do gabarito, restrições de precedência entre as tarefas e restrições de adjacência, ou seja:

$$\text{Min } t_F \tag{42}$$

$$t_j + p_j + q_j \leq t_F \quad j = 1, \dots, J \tag{43}$$

$$t_j \geq r_j \quad j = 1, \dots, J \tag{44}$$

$$t_j + p_j + q_j \leq d_j \quad j = 1, \dots, J \tag{45}$$

$$t_k + M(1 - y_{jk}) \geq t_j + p_j \quad \text{para todo par } (j, k) \in W \tag{46}$$

$$t_j + My_{jk} \geq t_k + p_k \quad \text{para todo par } (j, k) \in W \tag{47}$$

$$t_k \geq t_j + p_j + q_j \quad \text{para todo par } (j, k) \in A \tag{48}$$

$$t_k + M(1 - y_{jk}) \geq t_j + p_j \quad \text{para todo par } (j, k) \in B \tag{49}$$

$$t_j + My_{jk} \geq t_k + p_k \quad \text{para todo par } (j, k) \in B \tag{50}$$

$$t_j \geq 0, \quad j = 1, \dots, J, \quad y_{jk} \in \{0,1\}, \quad j = 1, \dots, J, \quad k = 1, \dots, J, \quad j \neq k \tag{51}$$

A função objetivo (42) junto com a restrição (43) garante que o *makespan*  $t_F$  seja mínimo e igual ao instante de término da última tarefa programada. As restrições (44) garantem que cada tarefa  $j$  só possa ser programada após seu instante de disponibilidade  $r_j$  para produção. As restrições (45) garantem que cada tarefa  $j$  seja completada antes do seu prazo de entrega  $d_j$ . No caso de todas as tarefas  $j$  terem o mesmo prazo de entrega  $d$  (i. e.,  $d_j = d$ ), então a restrição (43) e a função objetivo (42) já garantem o atendimento aos prazos de entrega e as restrições (45) são redundantes. As restrições (46) e (47) 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 (49) e (50) 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 (48) 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 (49) e (50) 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 (51) referem-se ao domínio das variáveis.

Observe que, por meio do modelo (42)-(51), 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 responsável pela produção no planejamento de expansão de capacidade e recursos humanos nos gabaritos em casos como, por exemplo, aumentos das vendas. Como a mão de obra é considerada ilimitada, as restrições de adjacência e tecnologia do gabarito automaticamente limitam o número de tarefas em montagem no gabarito, mas o número de tarefas em montagem nas bancadas é ilimitado.

### 4.3 Modelo matemático da fase 2

Para representar as restrições de adjacência, foi realizada uma adaptação na utilização dos recursos. Cada posto de trabalho do gabarito é representado como dois recursos. Assim, cada tarefa, quando precisa ser executada em algum posto de trabalho específico do gabarito, vai utilizar as duas unidades de recurso do posto. Essa mesma tarefa vai utilizar então, uma unidade de cada posto de trabalho adjacente e assim, impedir que outra tarefa possa utilizar postos de trabalho adjacentes durante a sua execução. A figura 40 ilustra as relações e usos de recursos de duas tarefas que utilizam as estações 2 e 4 do gabarito. A tarefa 1 utiliza os dois recursos do posto de trabalho 2, e mais um recurso de cada um dos postos de trabalho 1 e 3, e assim, impede que esses postos 1 e 3 sejam utilizados por uma outra tarefa durante a sua execução. A tarefa 2 utiliza os dois recursos do posto de trabalho 4,

e mais um recurso de cada um dos postos de trabalho 3 e 5. Perceba que os recursos do posto de trabalho 3 são divididos entre as duas tarefas, pois o posto 3 é adjacente aos postos 2 e 4. Assim, os postos de trabalho 1, 3 e 5 estão bloqueados por restrições de adjacência, enquanto as tarefas 1 e 2 estiverem sendo executadas no gabarito. O custo unitário da mão de obra,  $v_w$ , é considerado igual para todos os tipos de mão de obra  $w$  nas fases 2, 3 e 4.

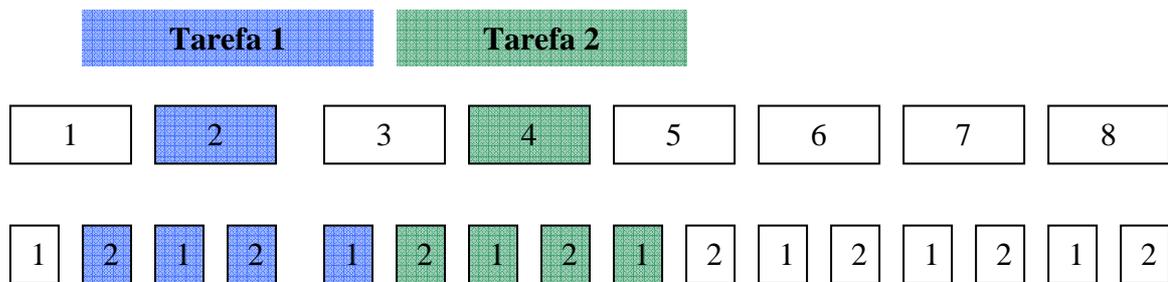


Figura 40: Representação dos recursos e utilização dos postos de trabalho do gabarito

Na fase 2, os operadores são divididos em dois grupos. Os especializados em montagens dos gabaritos e os especializados em montagens das bancadas. Assim, o modelo apresentado a seguir é inspirado em problemas de programação de projetos com restrições de recurso. Nesse caso especificamente, o problema é uma variação do TCPSP (18)-(22) apresentado na seção 3.4. A quantidade de recursos a ser utilizada é variável, mas o *makespan* é restrito, ou seja, previamente conhecido. O objetivo é minimizar a quantidade total de mão de obra utilizada para realizar as montagens, respeitando um dado *makespan*. Então, o objetivo é determinar qual o tamanho de cada um dos dois grupos de operadores de modo a atender todas as restrições. A definição da quantidade de operadores especializados em montagens de gabarito é essencial, pois, esse tipo de operador demora mais tempo para ser formado. Em geral, são os operadores que participaram do desenvolvimento do processo de produção durante a fase 1, de produção dos protótipos. Por isso, entre outros fatores, precisa-se ter definida uma cadência de produção com um horizonte de tempo de alguns meses desde o início da fase 1, durante a produção dos protótipos, com a finalidade de saber já na fase 2 qual seria a quantidade de operadores especializados em montagem de gabaritos necessária para atender à cadência de produção em série. Os operadores de bancada são formados mais rapidamente e não são tão críticos quanto os especializados nas operações de gabarito, embora sua formação também exija um tempo. Na fase 2, várias mudanças desenvolvidas no processo de produção da fase 1 acabam modificando os tempos de cada operação, reduzindo-os ou até eliminando operações.

No TCPSP, o tempo é discreto e o horizonte de tempo do programa de produção deve ser previamente definido como um parâmetro do modelo. No caso desse estudo, o tempo total,  $T$ , é dado pelo tempo de ciclo de cada avião, conforme a estratégia de produção por avião. Os tipos de recursos utilizados variam de 1 até  $K$ ; estes são os postos de trabalho do gabarito. No TCPSP, não é necessário diferenciar tarefa de operação, portanto, cada peça é composta apenas pelas operações  $j$ , diferentemente do modelo fase 1, em que duas operações formavam uma tarefa. Os parâmetros do modelo são:

$j$	operação, $j = 1, \dots, J$
$t$	tempo $t = 1, \dots, T$
$k$	tipo de recursos, $k = 1, \dots, K$
$w$	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 do recurso $k$ necessário para executar a tarefa $j$
$c_k$	quantidade disponível do recurso $k$
$p_j$	duração da operação $j$
$mo_{jr}$	quantidade de mão de obra do tipo $r$ 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 tarefa  $(h, j)$ , sendo  $h \neq j$ .

$H = \{(h, j) \mid \text{a tarefa } h \text{ precede a tarefa } j\}$ .

$G = \{ (j) \mid \text{a operação } j \text{ é executada no gabarito} \}$

As variáveis de decisão 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  $r$  a ser disponibilizada

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

$a_1$  quantidade de mão de obra de gabarito a ser disponibilizada

$a_2$  quantidade de mão de obra de bancada a ser disponibilizada

Então, o modelo matemático para a fase 2 é:

$$\text{Min} \sum_{r=1}^R v_w a_w \quad (52)$$

$$\sum_{t=1}^T x_{jt} = 1 \quad \text{para } \forall j \quad (53)$$

$$\sum_{t=r_h}^{d_h} t.x_{ht} \leq \sum_{t=r_j}^{d_j} (t-p_j).x_{jt} \quad \text{para } \forall (h, j) \in H \quad (54)$$

$$\sum_{j \in G} \sum_{b=t}^{t+p_j-1} rec_{jk} x_{jb} \leq c_k \quad \text{para } \forall (k, t) \quad (55)$$

$$\sum_{j=1}^J \sum_{b=t}^{t+p_j-1} mo_{jr} x_{jb} \leq a_w \quad \text{para } \forall w \quad (56)$$

$$x_{jt} = 0 \quad \begin{array}{l} t = 1, \dots, r_j + p_j - 2 \\ t = d_j + 1, \dots, T \\ \text{para } \forall j \end{array} \quad (57)$$

$$x_{jt} \in \{0, 1\}; a_l \in \mathbb{Z}^+, j = 1, \dots, J; t = 1, \dots, T; w = 1, \dots, W; k = 1, \dots, K \quad (58)$$

A função objetivo (52) é minimizar o custo de mão de obra utilizada. A restrição (53) garante que cada tarefa seja alocada uma e somente uma vez por todo o horizonte de planejamento. A restrição (54) garante o respeito às relações de precedência entre as tarefas. A restrição (55) garante o respeito à quantidade de recursos disponíveis, sendo que  $b$  é o momento de término da tarefa  $j$ . As restrições (56) garantem o respeito à quantidade de mão de obra. As restrições (57) garantem o respeito à data de entrega e data de chegada das tarefas. A restrição (58) é referente ao domínio das variáveis de decisão.

#### 4.4 Modelo matemático da fase 3

Na fase 3, os operadores que trabalham no gabarito também podem montar as operações nas bancadas, mas o inverso ainda não é possível. Nessa fase, os operadores de gabarito já ensinaram aos operadores de bancada, a mão de obra não especializada, todas as suas operações de montagem em bancadas. Então, a mão de obra especializada pode trabalhar em ambas as montagens (de gabarito e de bancada), porque não precisam mais utilizar o tempo que sobrava quando era dedicada apenas às montagens do gabarito para ensinar a mão de obra não especializada. Então, nesse caso as tarefas das bancadas podem ser executadas de duas formas. A primeira é utilizando a mão de obra do gabarito,  $a_1$ . A segunda é utilizando a mão de obra das bancadas,  $a_2$ . Então, na fase 3 é utilizado um modelo de programação de projetos com restrições de tempo e recursos e múltiplos modos de execução, inspirado no modelo (28)-(38), apresentado na seção 3.5 e baseado em Yamashita e Morabito (2007a).

O objetivo dessa fase continua a ser minimizar a quantidade total de mão de obra utilizada na montagem, porém, explorando a flexibilidade do grupo de operadores que podem trabalhar tanto nas montagens de gabarito quanto nas montagens de bancada. Com

essa nova capacidade, se o número e a duração das operações, a quantidade de gabaritos, os momentos de entrega e disponibilidade das tarefas são iguais à fase anterior, a solução da fase 2 é um limitante superior da fase 3.

Considere os parâmetros:

- $j$  operação,  $j = 1, \dots, J$   
 $i$  modo de execução,  $i = 1, \dots, I$   
 $t$  tempo de término  $t = 1, \dots, T$   
 $k$  tipos de recursos,  $k = 1, \dots, K$   
 $w$  tipo de mão de obra,  $w = 1, \dots, W$   
 $r_j$  tempo em que a tarefa  $j$  está disponível, onde  $1 \leq r_j \leq T$  e inteiro  
 $d_j$  tempo de entrega da tarefa  $j$ , onde  $1 \leq d_j \leq T$  e inteiro  
 $v_w$  custo unitário da mão de obra do tipo  $w$   
 $rec_{iki}$  quantidade de recursos necessários para executar a tarefa  $j$  no modo de operação  $i$   
 $p_i$  duração da operação  $j$   
 $mog_{jw}$  quantidade de mão de obra do tipo  $w$  utilizada para executar a tarefa  $j$  no gabarito  
 $mob_{jiw}$  quantidade de mão de obra do tipo  $w$  utilizada para executar a tarefa  $j$  no modo de execução  $i$  na bancada  
 $H = \{ (h, j) \mid \text{a tarefa } h \text{ precede a tarefa } j \}$   
 $G = \{ (j) \mid \text{a operação } j \text{ é executada no gabarito} \}$

As variáveis de decisão são:

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

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

Sendo que, neste caso da fase 3 ( $L = 2$ ):

$a_1$  quantidade de mão de obra de gabarito a ser disponibilizada

$a_2$  quantidade de mão de obra de bancada a ser disponibilizada

O modelo matemático é:

$$\text{Min} \sum_{l=1}^L v_w a_w \quad (59)$$

$$\sum_{i=1}^I \sum_{t=1}^T x_{jti} = 1 \quad \text{para } \forall j \quad (60)$$

$$\sum_{i=1}^I \sum_{r_h}^{d_h} t \cdot x_{hti} \leq \sum_{i=1}^I \sum_{r_j}^{d_j} (t - p_j) \cdot x_{jti} \quad \text{para } \forall (h, j) \in H \quad (61)$$

$$\sum_{i=1}^I \sum_{j \in G} \sum_{b=t}^{t+p_j-1} rec_{jki} x_{jbi} \leq c_k \quad \text{para } \forall (k, t) \quad (62)$$

$$\sum_{j \in G} \sum_{b=t}^{t+p_j-1} mog_{jw} x_{jbi} + \sum_{j \in J \setminus G} \sum_{b=t}^{t+p_j-1} mob_{jwi} x_{jbi} \leq a_w \quad \text{para } i = 1 \text{ e } w = 1 \quad (63)$$

$$\sum_{j \in J \setminus G} \sum_{b=t}^{t+p_j-1} mob_{jwi} x_{jbi} \leq a_w \quad \text{para } i = 2 \text{ e } w = 2 \quad (64)$$

$$x_{jti} = 0 \quad \begin{array}{l} t = 1, \dots, r_j + p_j - 2 \\ t = d_j + 1, \dots, T \\ \forall j \\ \forall i \end{array} \quad (65)$$

$$x_{jti} = 0 \quad j \in G, i = 2, \forall t \quad (66)$$

$$x_{jti} \in \{0, 1\}; a_l \in \mathbb{Z}^+, j = 1, \dots, J; t = 1, \dots, T; w = 1, \dots, W; k = 1, \dots, K \quad (67)$$

A função objetivo (59) é minimizar o custo total de mão de obra utilizada. As restrições (60)-(67) têm as mesmas finalidades do modelo apresentado na seção 4.3 porém, adaptados para múltiplos modos de execução.

É importante observar que o modelo (59)-(67) pode ser redefinido como um modelo de programação de projetos com apenas um único modo de execução (i. e.,  $I = 1$ ). Para isso, é necessário reescrever as restrições (63) e (64), utilizando uma variável de folga,  $y_t$ , que representa a folga que a mão de obra especializada tem a cada tempo  $t$ , e que pode ser utilizada nas montagens de bancada. Considere os mesmos parâmetros do modelo (59)-(67), mas agora as variáveis de decisão são  $x_{jt}$  (o invés de  $x_{jti}$ ). Então, o modelo é reescrito como:

$$\text{Min} \sum_{l=1}^L v_w a_w \quad (68)$$

$$\sum_{t=1}^T x_{jt} = 1 \quad \text{para } \forall j \quad (69)$$

$$\sum_{r_h}^{d_h} t \cdot x_{ht} \leq \sum_{r_j}^{d_j} (t - p_j) \cdot x_{jt} \quad \text{para } \forall (h, j) \in H \quad (70)$$

$$\sum_{j \in G} \sum_{b=t}^{t+p_j-1} rec_{jk} x_{jb} \leq c_k \quad \text{para } \forall (k, t) \quad (71)$$

$$\sum_{j \in G} \sum_{b=t}^{t+p_j-1} mog_{jw} x_{jb} = a_w - y_t \quad w = 1 \quad (72)$$

$$\sum_{j \in J \setminus G} \sum_{b=t}^{t+p_j-1} \text{mob}_{jw} x_{jb} \leq a_w + y_t \quad w = 2 \quad (73)$$

$$x_{jt} = 0 \quad \begin{array}{l} t = 1, \dots, r_j + p_j - 2 \\ t = d_j + 1, \dots, T \\ \text{para } \forall j \end{array} \quad (74)$$

$$x_{jt} \in \{0, 1\}; a_l \in \mathbb{Z}^+, y_t \geq 0, j = 1, \dots, J; t = 1, \dots, T; w = 1, \dots, W; k = 1, \dots, K \quad (75)$$

As restrições (69)-(75) têm as mesmas finalidades das restrições (60)-(67), porém, com apenas um único modo de execução. No modelo (68)-(75), como só existe um único modo de execução, então, a quantidade de variáveis  $x_{jt}$  é muito menor que no modelo (60)-(67), que possui variáveis  $x_{jit}$ . Assim, espera-se que o modelo (69)-(75) tenha um desempenho computacional melhor que o modelo (60)-(67), conforme é apresentado na seção 5.1.3. Repare que nas restrições (72) e (73),  $x_{jb}$ ,  $a_w$ ,  $\text{mog}_{jw}$  e  $\text{mob}_{jw}$  são inteiros, logo,  $y_t$  pode ser um número real positivo, pois, as restrições (72) e (73) garantem que  $y_t$  seja inteiro.

É importante ressaltar que, caso a mão de obra especializada não seja limitada, a solução ótima poderá conter apenas operadores especializados, o que permite minimizar ainda mais a quantidade de mão de obra. Isso significa que a solução ótima pode sugerir uma equipe formada apenas por operadores especializados que podem montar tanto no gabarito quanto nas bancadas. Isto é uma característica da fase 4 e só pode ser considerada na fase 3, caso a quantidade de mão de obra de gabarito disponível na prática seja suficiente para atender à solução do modelo. Caso contrário, deve-se introduzir uma restrição adicional, que limita a quantidade do recurso  $a_l$ , obrigando a solução a conter mão de obra não especializada, ou seja, de bancada. Uma outra forma de restringir a equipe de mão de obra especializada seria utilizando custos diferentes na função objetivo (59) ou (68) para cada uma das variáveis  $a_w$ , ou seja, um peso maior para a mão de obra de gabarito. Entretanto, essa maneira pode levar a solução ótima a subutilizar a mão de obra de gabarito e usar maior quantidade da mão de obra da bancada. Ou seja, não utilizar todos operadores especializados que existem na prática, mas utilizar uma quantidade maior de operadores não especializados, diminuindo a qualidade da solução.

#### 4.5 Modelo matemático da 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 no gabarito e então, a equipe é formada somente por operadores totalmente flexíveis. O objetivo

é minimizar o tamanho da equipe, explorando a flexibilidade de todos. A solução ótima do modelo matemático (52)-(58) da seção 4.4, 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 fase 4. O modelo matemático da fase 4 é igual ao modelo (52)-(58) da fase 2, com a diferença que não existem dois tipos de mão de obra, mas apenas um tipo (i. e.  $W = 1$ ).

A fase 4 é a última fase da curva de aprendizagem, quando a área de montagem do subconjunto atingiu a sua maturidade. Assim, durante o planejamento da produção de um modelo de avião novo, a fase 4 seria o período em que a curva de aprendizagem pode ser considerada estável e melhorias incrementais iriam exigir grandes esforços. O tamanho da equipe nessa fase é um importante objetivo a ser otimizado, pois é nela também que costuma ocorrer a produção do avião em maiores cadências.

## 5 Resultados Computacionais

Os modelos apresentados no capítulo 4 foram implementados na linguagem de modelagem GAMS 23.0 e resolvidos pelo *software* CPLEX 11.2.1.0 em um computador com 12 GB de RAM e processador Intel i7 2.80 GHz. Em todas as instâncias e modelos, o número de núcleos que o CPLEX poderia utilizar foi configurado para 4, para o máximo aproveitamento da capacidade de processamento, e o tempo máximo de processamento foi fixado em 10 horas ou 600 minutos.

### 5.1 Resultados computacionais da montagem do subconjunto

#### 5.1.1 Exemplo computacional da fase 1

Na fase 1, como apresentado anteriormente, a mão de obra necessária pode ser considerada ilimitada, pois esta é a fase de projeto e produção de protótipos e, caso seja necessário, pode-se contratar mão de obra ou realocar de outros setores da empresa. No entanto, isso não significa que o processo de produção não possa mudar ao longo da fase 1, sem perder as características dessa fase. Nesse capítulo, são apresentados dois exemplos de processos de produção ligeiramente diferentes, com variações no número e duração das operações.

A tabela 5, no apêndice A, apresenta as durações de cada operação de montagem do exemplo 1. As durações apresentadas não são as reais, pois a unidade de tempo original foi modificada e chamada apenas de unidade de tempo (u. t.), com a finalidade de proteger os interesses da empresa. A rede de precedência das tarefas da tabela 5 é similar à rede apresentada na figura 31, com a diferença que as peças 4 e 5 do subconjunto são montadas em três etapas, conforme a rede de precedência apresentada na figura 41. Nesse caso, são apresentadas as tarefas de um subconjunto completo, sendo que outro subconjunto idêntico forma 1 avião, no total de 36 tarefas.

Aplicando o modelo (42)-(51) à rede apresentada na figura 41 do exemplo 1, pode-se encontrar o menor *makespan* para a rede, considerando a mão de obra um recurso ilimitado. O modelo possui 464 variáveis contínuas, 426 variáveis binárias e 987 equações. O CPLEX demorou menos de 1 segundo para encontrar a solução ótima. A figura 42 apresenta o programa de produção encontrado. A solução ótima é 195 u.t. e são necessárias 14 pessoas para realizar o programa de produção.

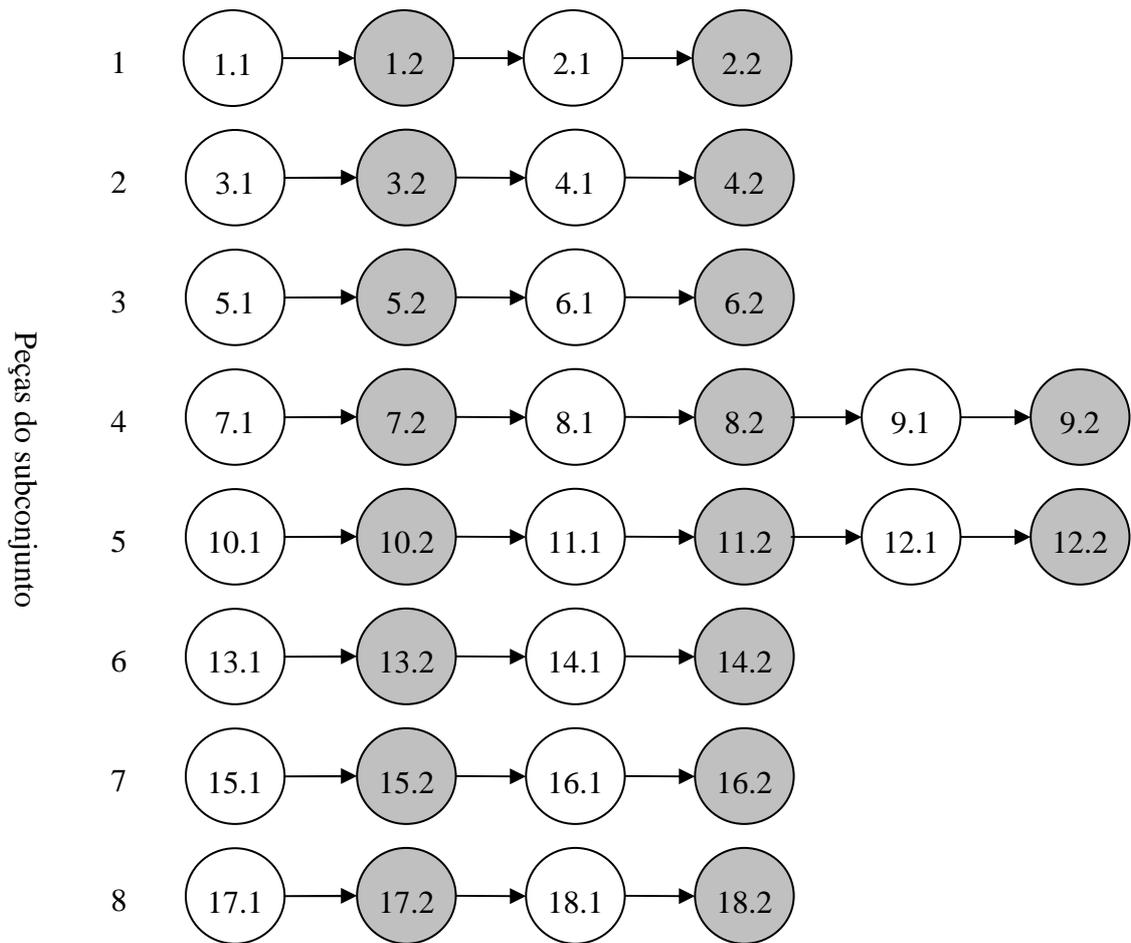


Figura 41: Rede de precedência das tarefas de montagem do subconjunto no exemplo 1

A solução apresentada não é nivelada, conforme pode ser observado na figura 43, em que é apresentada a variação da quantidade de mão de obra trabalhando em cada momento do programa de produção, separada em mão de obra de gabarito e mão de obra de bancada. Como é esperado, as operações de gabarito começam com o máximo possível de mão de obra trabalhando, que são 4 pessoas devido às restrições de adjacência, e declina até o fim do programa. A mão de obra de bancada é mais seriamente desnivelada, alcançando picos com 11 pessoas trabalhando ao mesmo tempo.

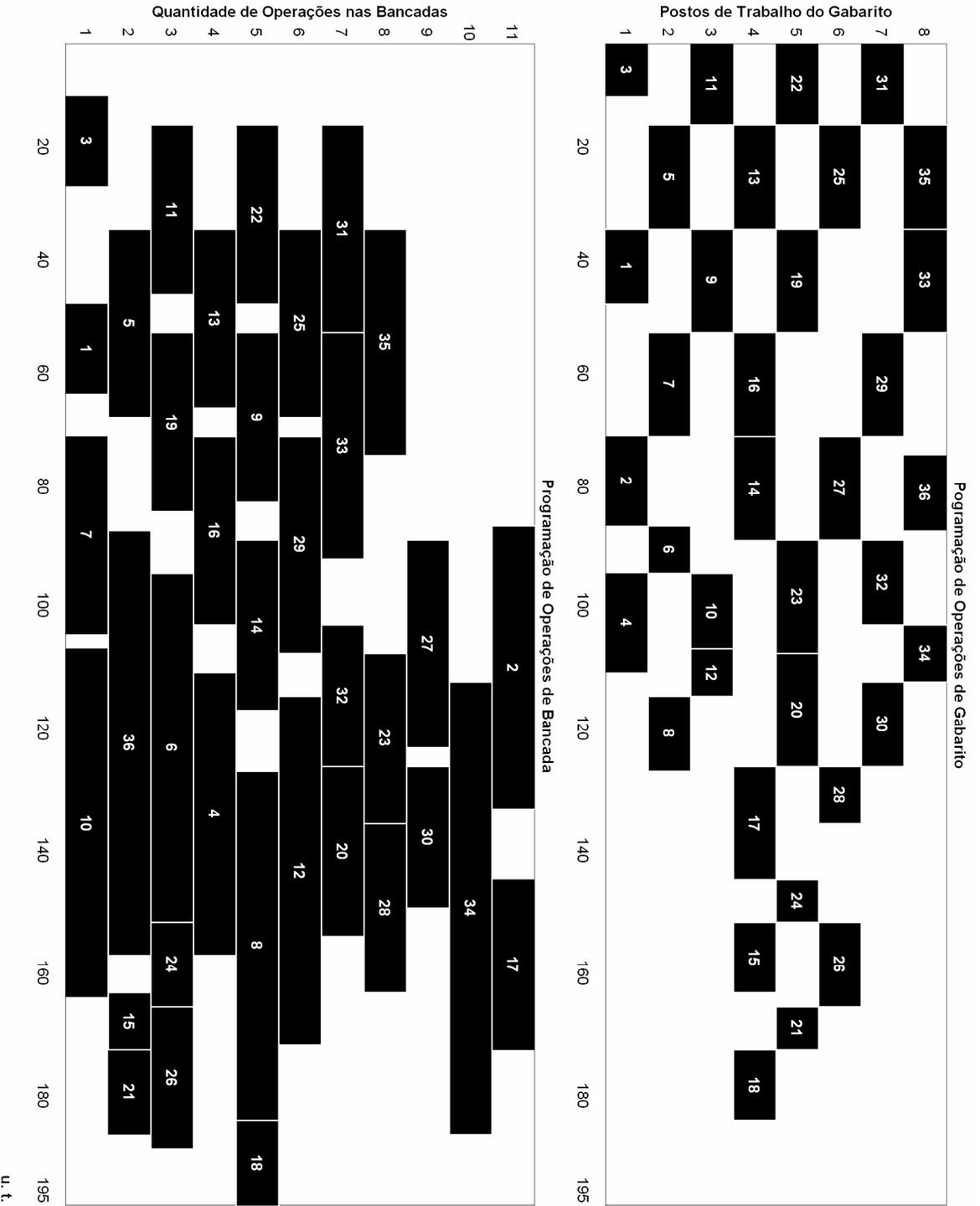


Figura 42: Gráfico de Gantt com o programa de produção do exemplo 1 na fase I do subconjunto com o modelo (42)-(51)

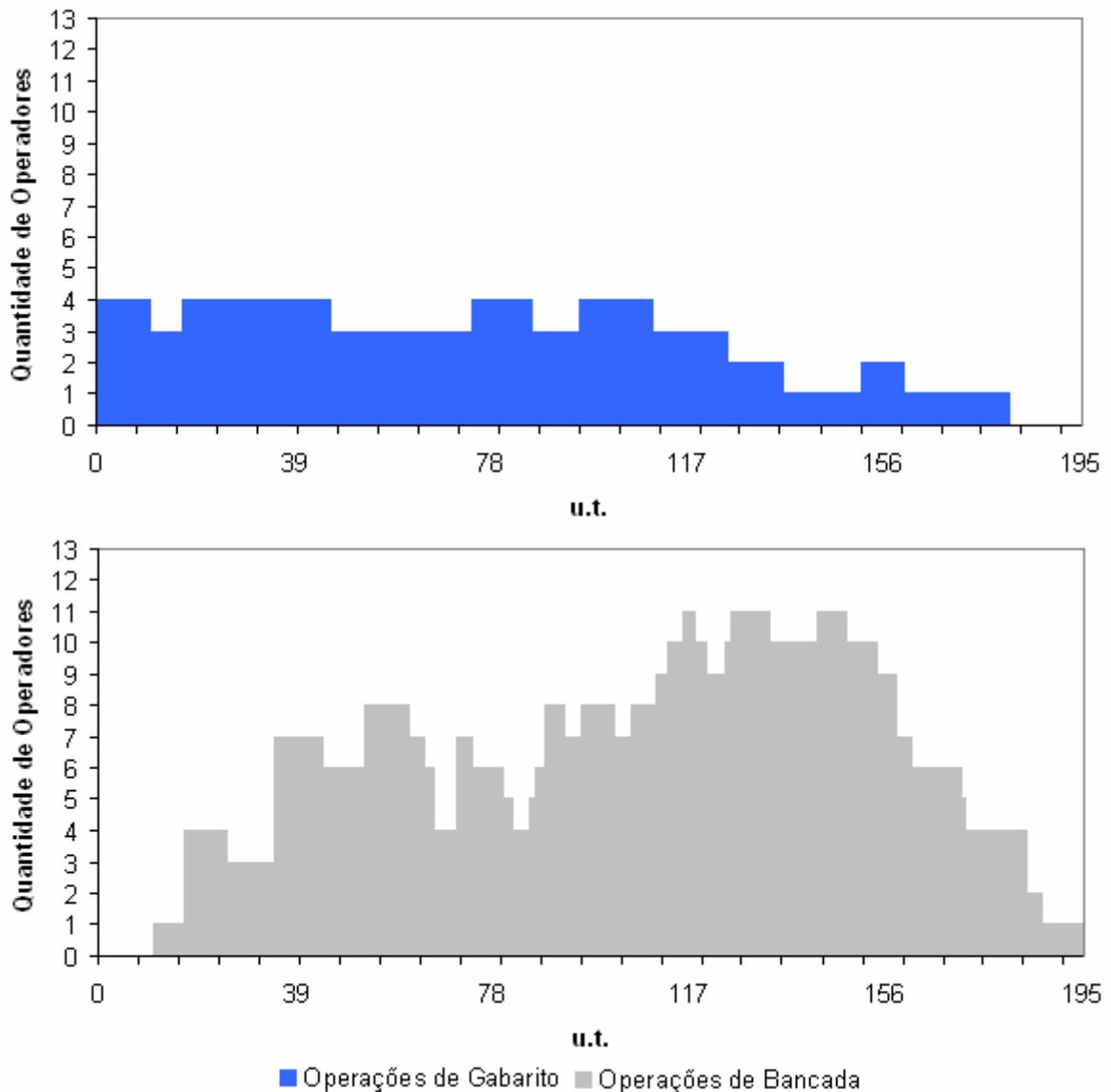


Figura 43: Variação do número o de operadores no gabarito e na bancada para a fase 1 do exemplo 1 utilizando o modelo (42)-(51)

A quantidade total de operadores necessários para realizar o programa de produção da figura 42 é apresentada na figura 44. Comparando-se a figura 43 com a figura 44 é possível observar que a variação dos dois tipos de mão de obra é maior individualmente do que a variação da mão de obra total, embora a quantidade total de mão de obra varie ao longo do programa, atingindo picos com 14 pessoas no momento 121 u. t.. Em média, são necessárias 9 pessoas, com desvio padrão de 3,28 pessoas.

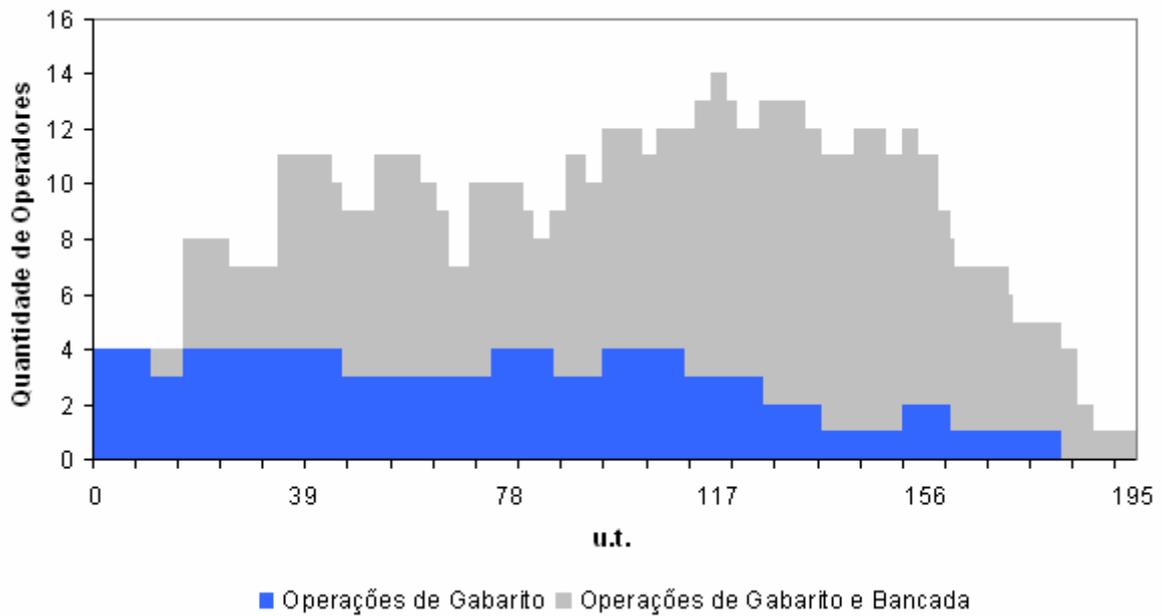


Figura 44: Variação do número de operações para o exemplo 1 com o modelo (42)-(51)

Com o *makespan* ótimo do problema, é possível utilizar os modelos de programação de projetos para nivelar os recursos, definindo o tempo do projeto como o *makespan* encontrado no modelo (42)-(51) e o objetivo de minimizar a quantidade de recursos necessários para realizar o projeto. Assim, para nivelar os recursos, foi utilizado o modelo (52)-(58), mas qualquer um dos outros modelos de programação de projetos das outras fases poderia ser utilizado para nivelar os recursos.

O modelo (52)-(58) aplicado ao exemplo 1 com o formato da fase 4, ou seja, todos os operadores podem realizar qualquer montagem, possui 10.046 variáveis binárias e 2.862 equações e demorou 154 minutos para encontrar a solução ótima. Com o *makespan* fixado em 195 u. t., o modelo (52)-(58) encontrou a solução ótima com 10 pessoas realizando a montagem, ou seja, a solução reduz em 4 pessoas a necessidade total de mão de obra encontrada na solução do modelo matemático (42)-(51), que era de 14. A figura 45 apresenta o programa de produção encontrado pela solução do modelo (52)-(58).

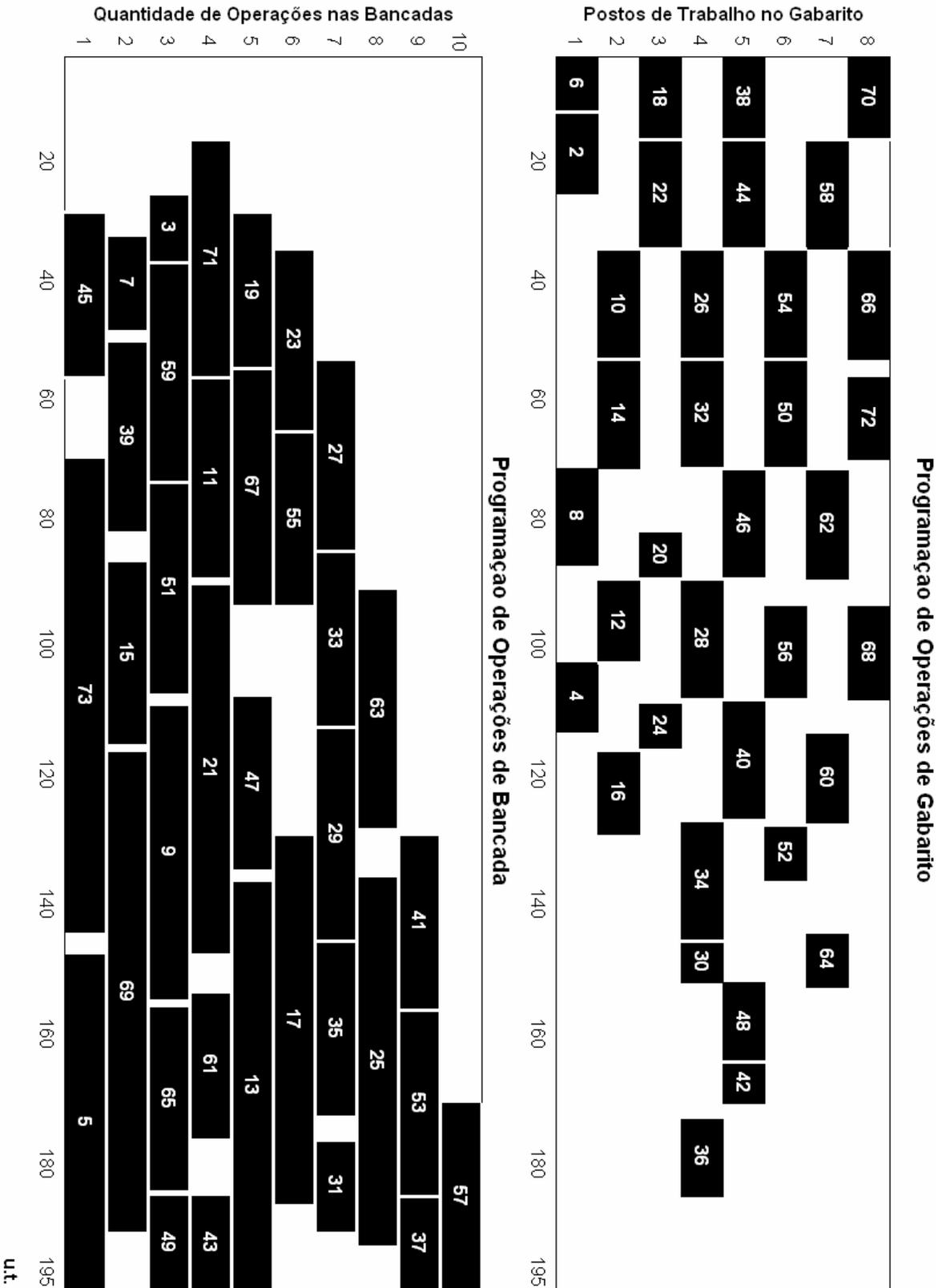


Figura 45: Gráfico de Gantt com o programa de produção do exemplo 1 na fase I do modelo (52)-(58)

A figura 46 apresenta a variação da quantidade de mão de obra para cada tipo de operação necessária para executar o programa da figura 45.

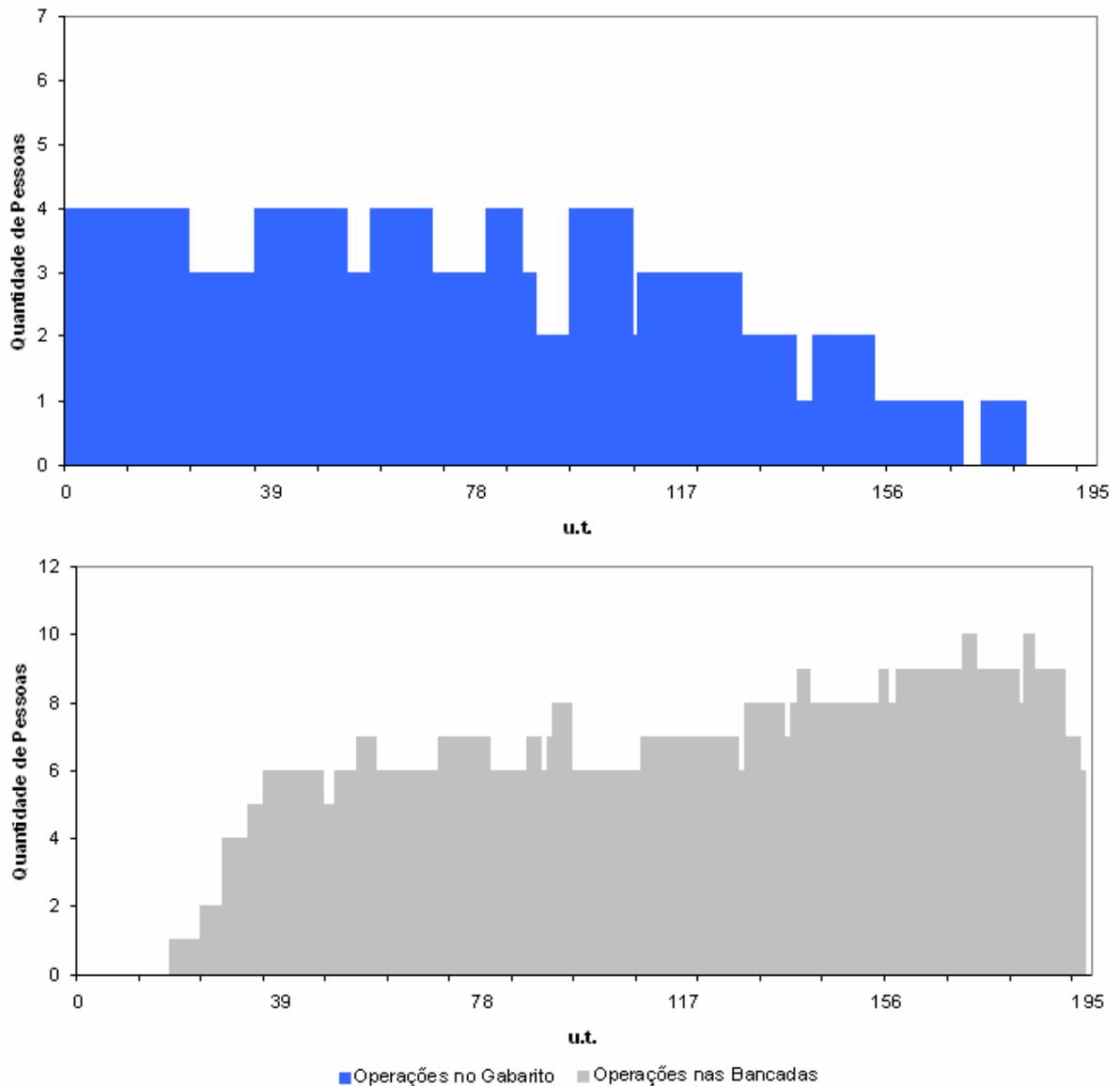


Figura 46: Variação do número de operadores no gabarito e na bancada para a fase 1 do exemplo 1 do modelo (52)-(58)

A figura 47 apresenta a variação do total de operadores na montagem do programa da figura 45. É possível observar que o número de operadores varia de 4 até 10 e que a variação da quantidade de pessoas trabalhando ao longo do programa de produção é menor que no caso da figura 44. Então, com o modelo (42)-(51) é possível determinar qual o mínimo *makespan* para executar o projeto e assim, determinar qual o menor tempo de ciclo e a capacidade do gabarito de montagem em questão. Com essa informação e utilizando o modelo (52)-(58), pode-se nivelar e minimizar a quantidade de mão de obra necessária para executar o projeto no menor *makespan*. Em média são necessárias 8,9 pessoas, com desvio

padrão de 2,02 pessoas, ou seja, o nivelamento diminuiu a variância do recurso ao longo do programa de produção.

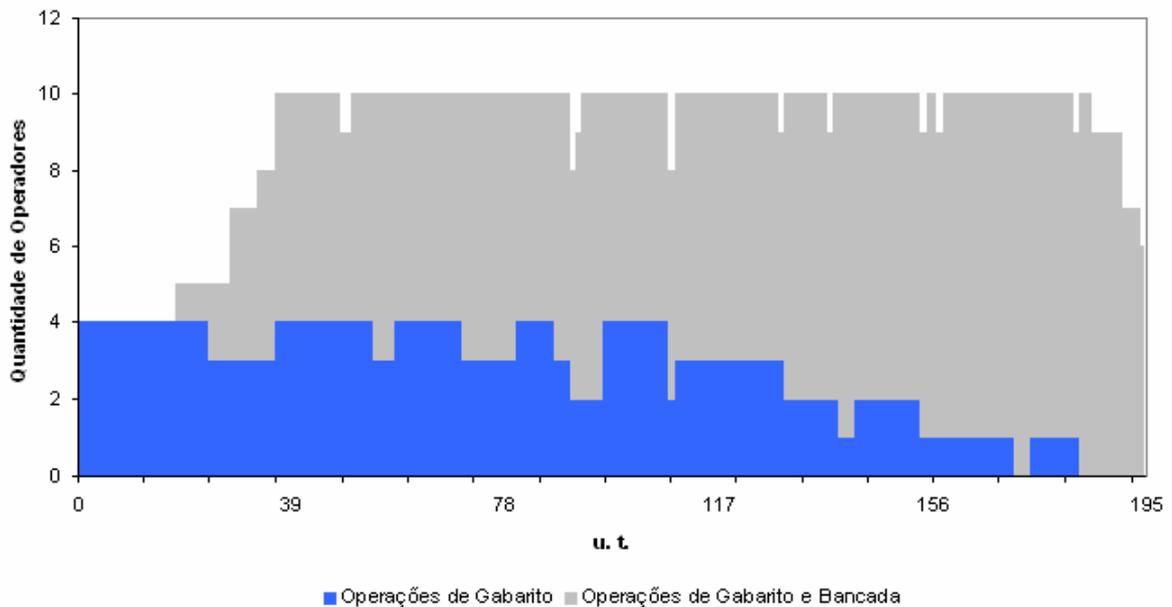


Figura 47: Variação do número de operadores para o exemplo 1 com o modelo (52)-(58)

Um segundo exemplo (exemplo 2) foi considerado para analisar o desempenho do modelo da fase 1. A empresa realizou algumas melhorias no processo de montagem modificando os tempos de tarefas, aumentando ou diminuindo os tempos de operação de algumas tarefas, modificando a seqüência de montagem e eliminando algumas operações. No exemplo 2, as peças 4 e 5 são montadas com apenas 2 tarefas, enquanto no exemplo 1 elas eram montadas por 3 tarefas. A peça 3 é montada por apenas 1 tarefa, enquanto no exemplo 1 ela precisava de 2 tarefas. Alguns tempos são diferentes, conforme é apresentado na tabela 6 no apêndice A. O número total de tarefas dos dois subconjuntos é 30, ou seja, 6 tarefas a menos que o exemplo 1, que era formado por 36 tarefas. A figura 48 apresenta a rede de precedência das tarefas do exemplo 2.

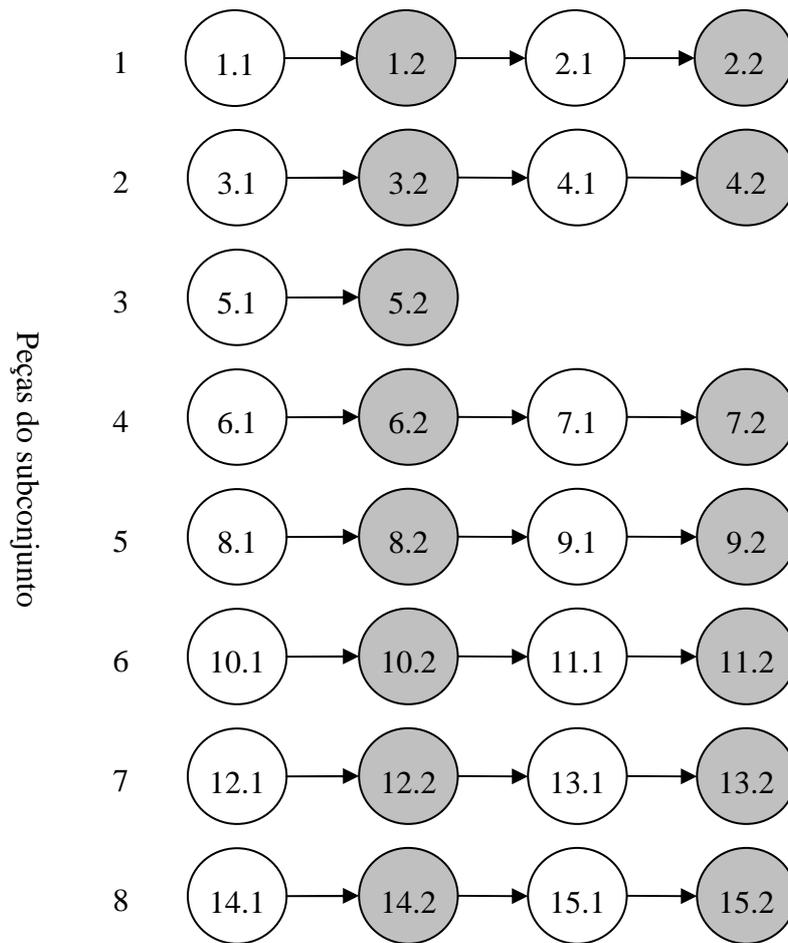


Figura 48: Rede de precedência das tarefas de montagem do subconjunto no exemplo 2

O modelo matemático (42)-(51) do exemplo 2 possui 310 variáveis contínuas, 278 variáveis binárias e 667 equações. O CPLEX demorou menos de 1 segundo para encontrar a solução ótima de 161 u.t., cerca de 16% menor que os 195 u.t. necessários para o exemplo 1. A figura 49 apresenta o programa de produção da solução ótima encontrada pelo modelo (42)-(51).

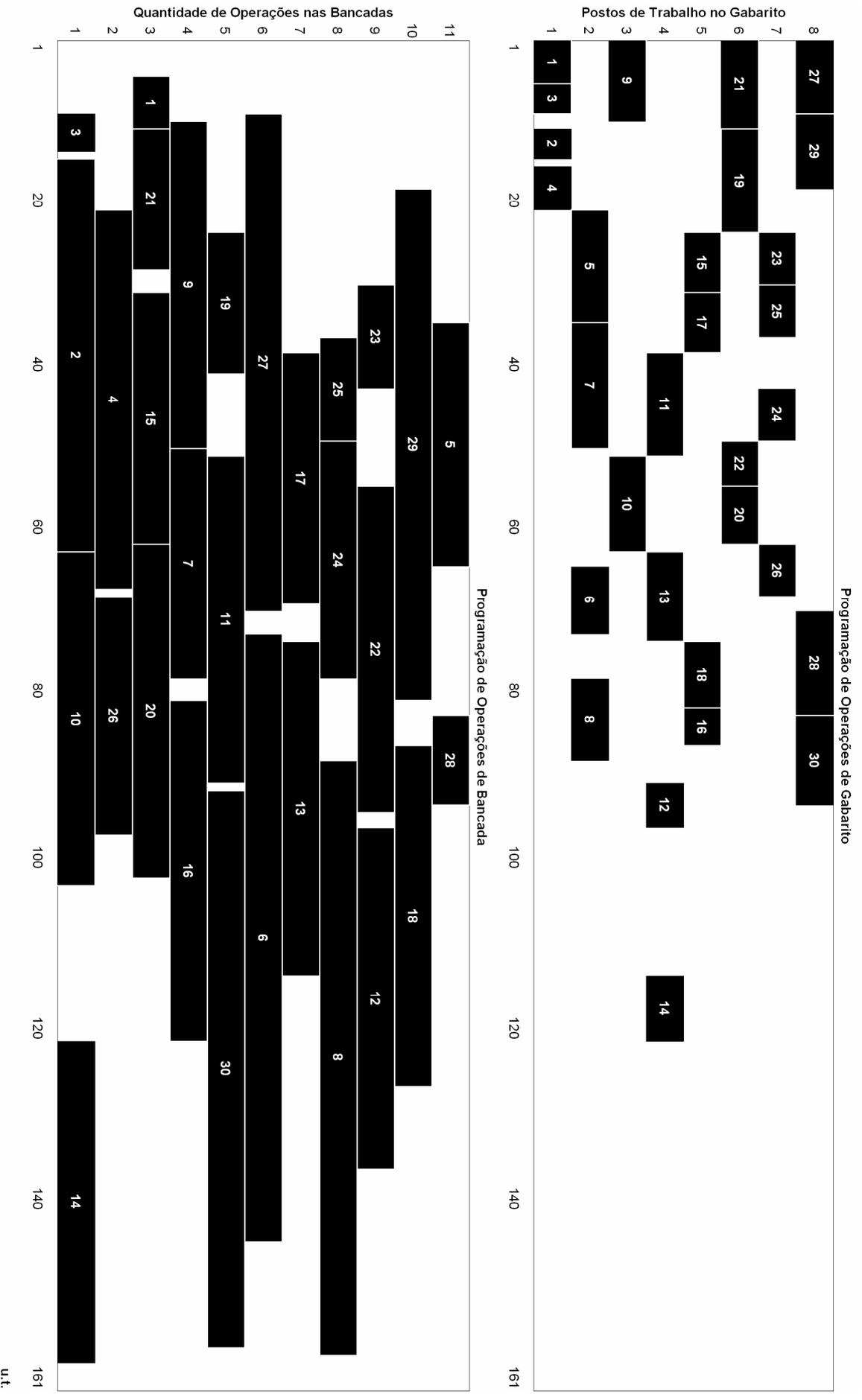


Figura 49: Gráfico de Ganti com o programa de produção do exemplo 1 na fase 1 do modelo (42)-(51)

u.t.

A solução ótima encontrada pelo modelo (42)-(51) precisa de 13 pessoas para executar o projeto da forma que a solução programou as tarefas. A figura 50 apresenta a variação de mão de obra para os dois tipos de operação. Como na figura 43, existe um desnivelamento da mão de obra, particularmente nas operações de bancada que em momentos como os 50 e 85 precisam de 11 pessoas e de 145 em diante precisa de 4 pessoas.

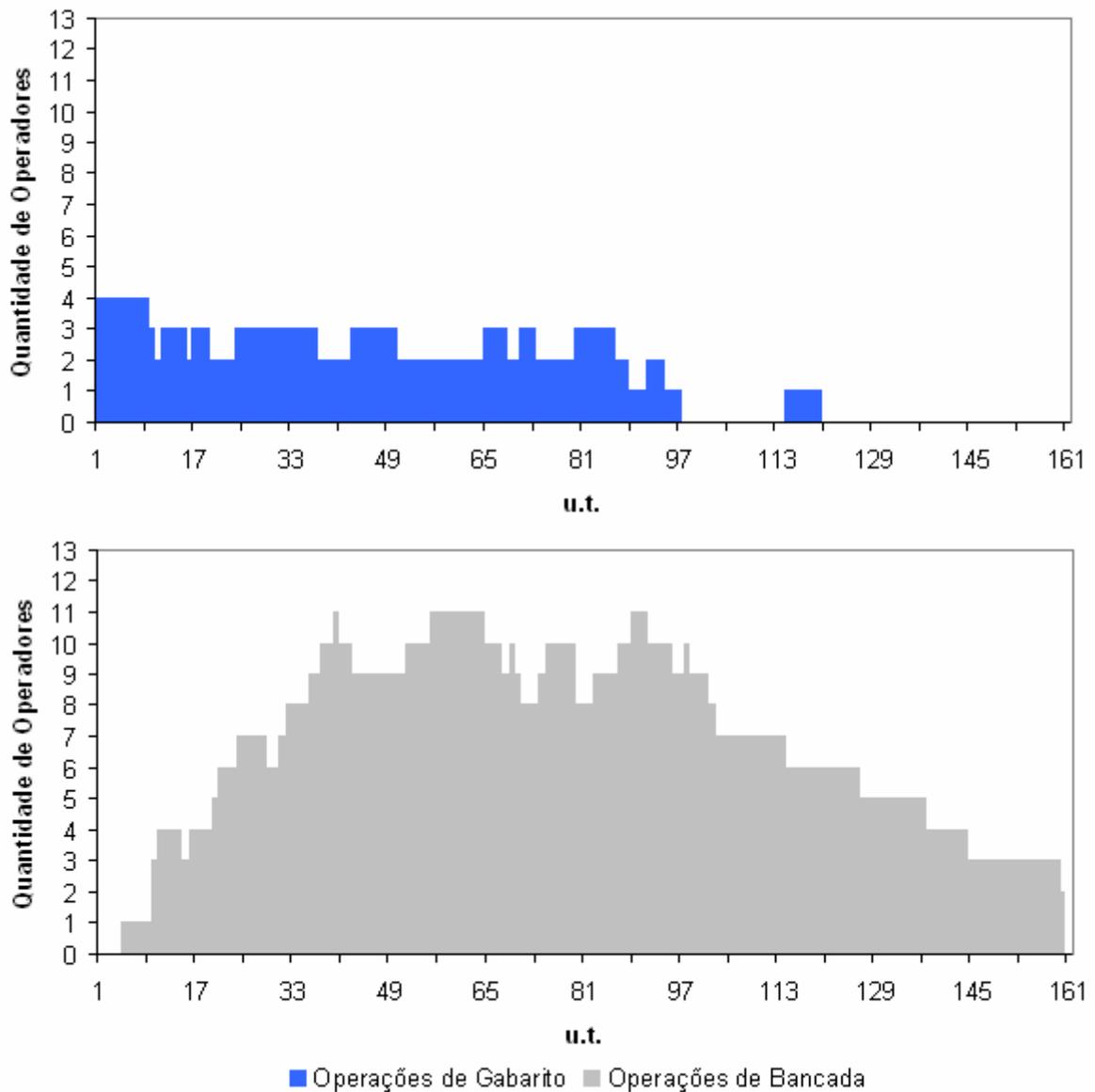


Figura 50: Variação do número de operadores no gabarito e na bancada para a fase 1 do exemplo 1 do modelo (42)-(51)

A figura 51 apresenta a variação da quantidade total de mão de obra durante o programa de produção. Entre 55 e 65 u. t., o programa precisa de 13 pessoas. Como no caso do exemplo 1, esse programa é muito desnivelado e não serve para ser implementado diretamente na produção. Em média, são necessárias 7,9 pessoas com desvio padrão de 3,54 pessoas.

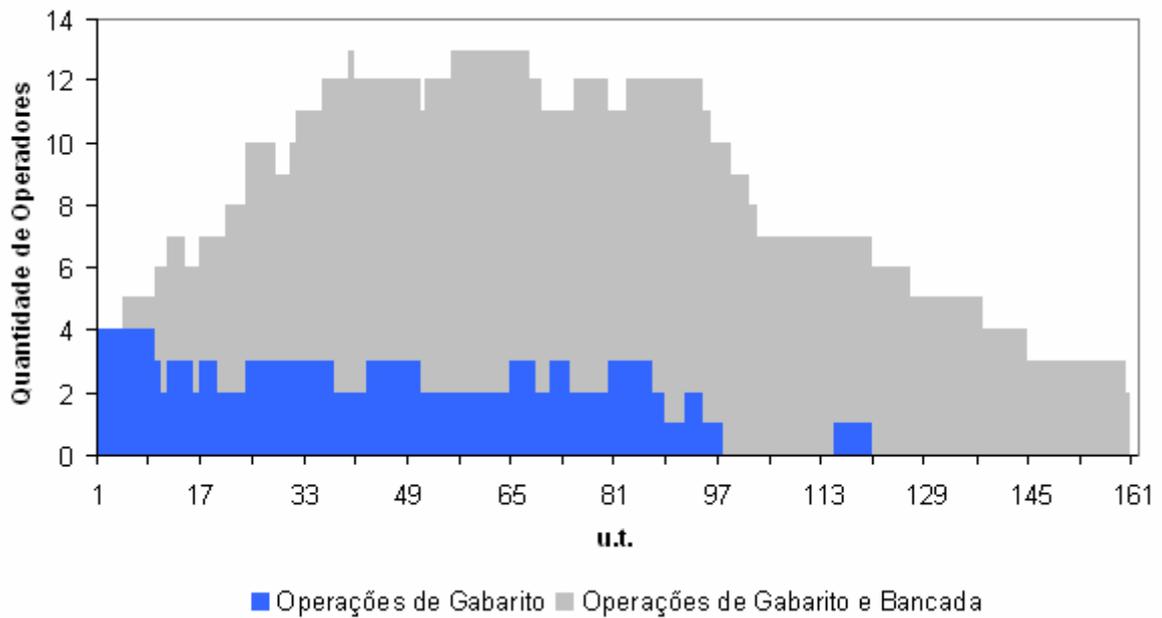


Figura 51: Variação do número de operadores para o exemplo 1 com o modelo (42)-(51)

Utilizando o *makespan* de 161 encontrado pelo modelo (42)-(51) e o modelo (52)-(58) da fase 4, é possível nivelar a mão de obra da solução. O modelo matemático (52)-(58) utilizado para nivelar a mão de obra, que é o modelo matemático da fase 4, tem 14.506 variáveis discretas e 3.524 equações. O CPLEX demorou 8 minutos para encontrar a solução ótima. A figura 52 apresenta o gráfico de Gantt com o programa de produção do exemplo 2.

O programa de produção da figura 52 também apresenta diversos momentos que alguns recursos ficam ociosos. O desvio padrão do número de operadores no exemplo 1, calculado em relação ao número médio de operadores trabalhando em cada momento, é de 3,54 pessoas e do exemplo 2 é de 1,09 pessoas. Isso mostra que as variações existem e no caso do exemplo 2 são maiores. A figura 53 apresenta a variação da quantidade de mão de obra para cada tipo de operação ao longo do programa. A solução ótima encontrada pelo modelo (52)-(58) precisa de 9 pessoas para executar o programa de produção, ou seja, 4 pessoas a menos que a solução do modelo (42)-(51), que precisa de 13 pessoas.

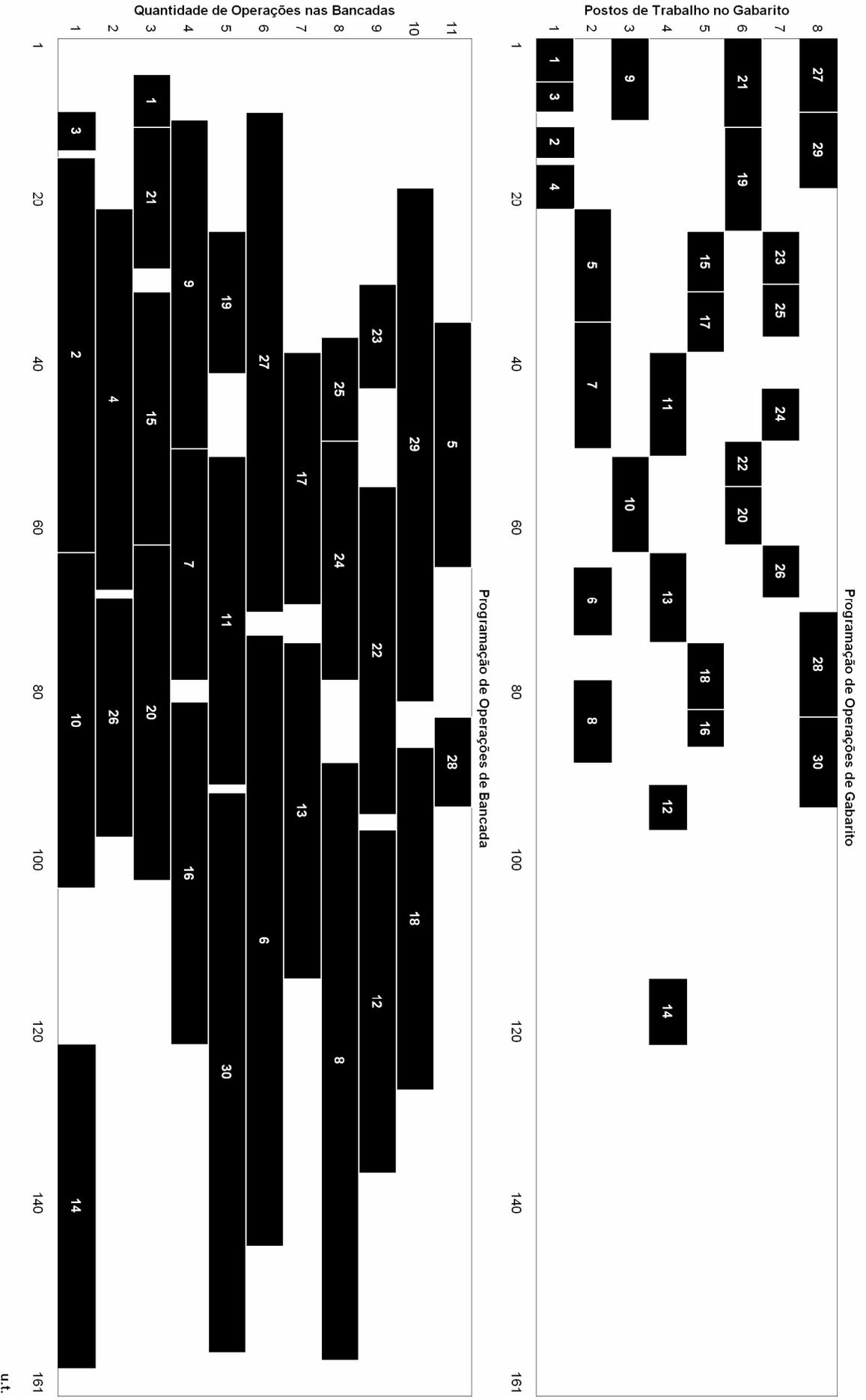


Figura 52: Gráfico de Ganti com o programa de produção do exemplo 2 na fase 1 do modelo (52)-(58)

u.t.

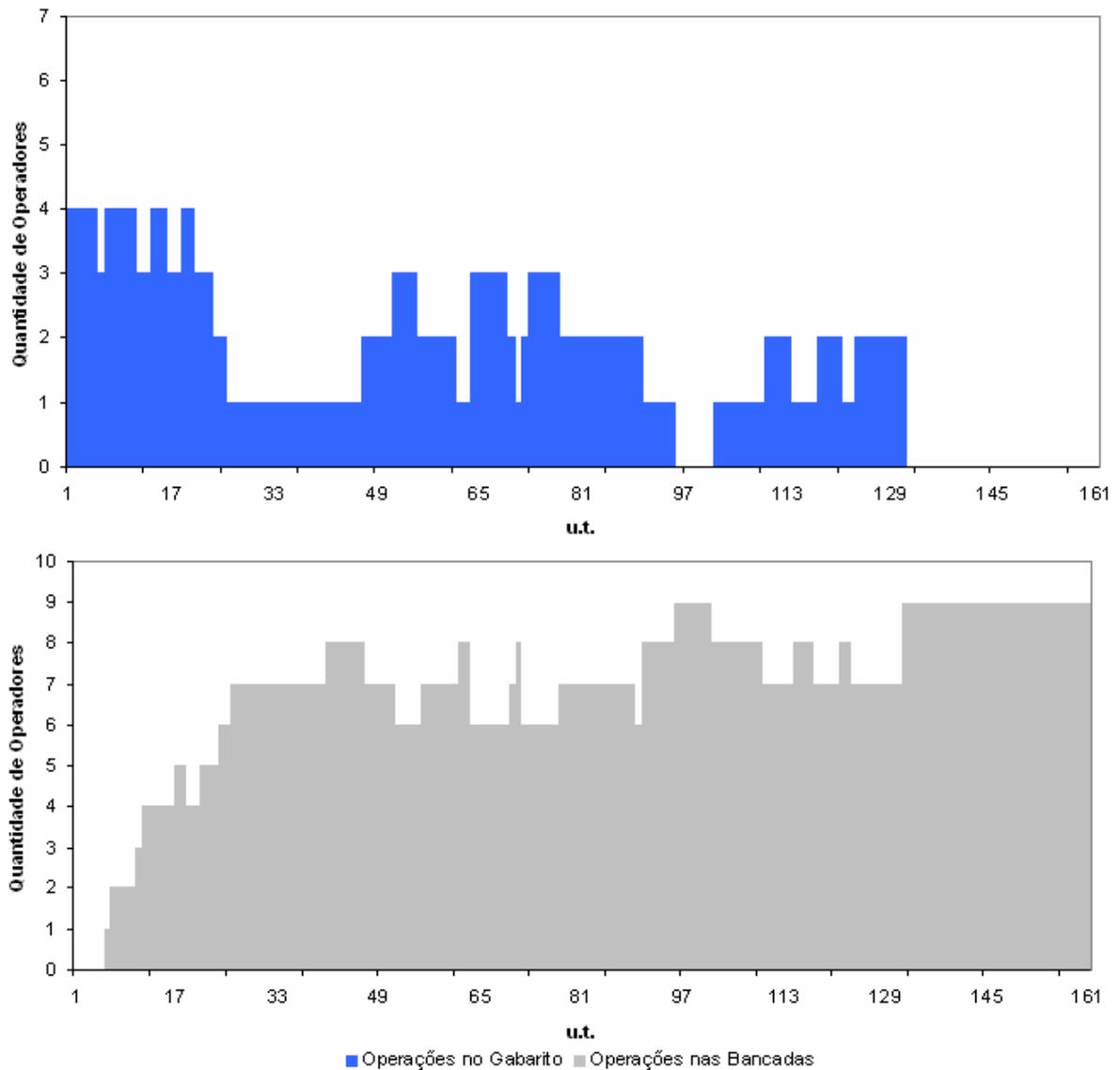


Figura 53: Variação do número de operadores no gabarito e na bancada para a fase 1 do exemplo 2 com o modelo (52)-(58)

A figura 54 apresenta a variação do número de operadores ao longo do programa de produção. É possível observar que o máximo de operadores no exemplo 2 é 9, ou seja, um operador a menos que no exemplo 1, em que foram necessários 10 operadores para executar o programa de produção. No exemplo 2, o recurso mão de obra também é melhor nivelado, conforme se pode observar ao comparar a figura 51 com a figura 54.

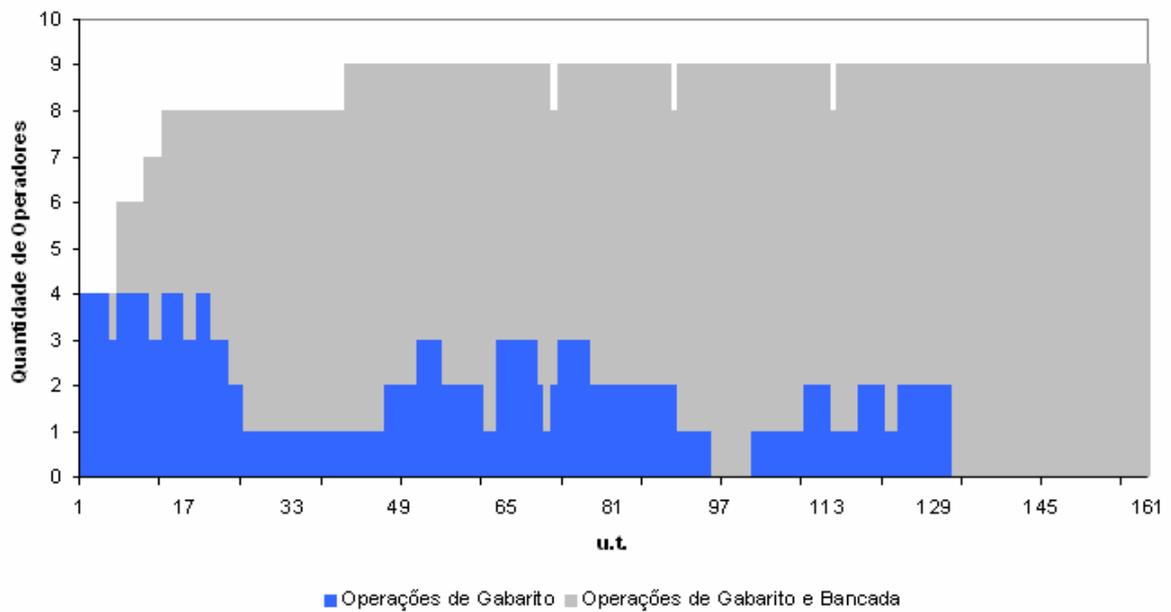


Figura 54: Variação do número de operadores no exemplo 2

As modificações do processo no exemplo 2 são exemplos de modificações na produção que acabam diminuindo a quantidade em homens-hora necessária para a produção de um avião e assim, percorrendo a curva de aprendizagem.

### 5.1.2 Exemplo computacional da fase 2

Os dados utilizados nesse exemplo são os mesmos do exemplo 2, que são apresentados na tabela 6 no apêndice A. O modelo matemático (52)-(68) tem 17.866 variáveis binárias, 8.295 equações e foram necessários 49 minutos para o CPLEX encontrar a solução ótima. A figura 55 ilustra o programa de produção da solução ótima.

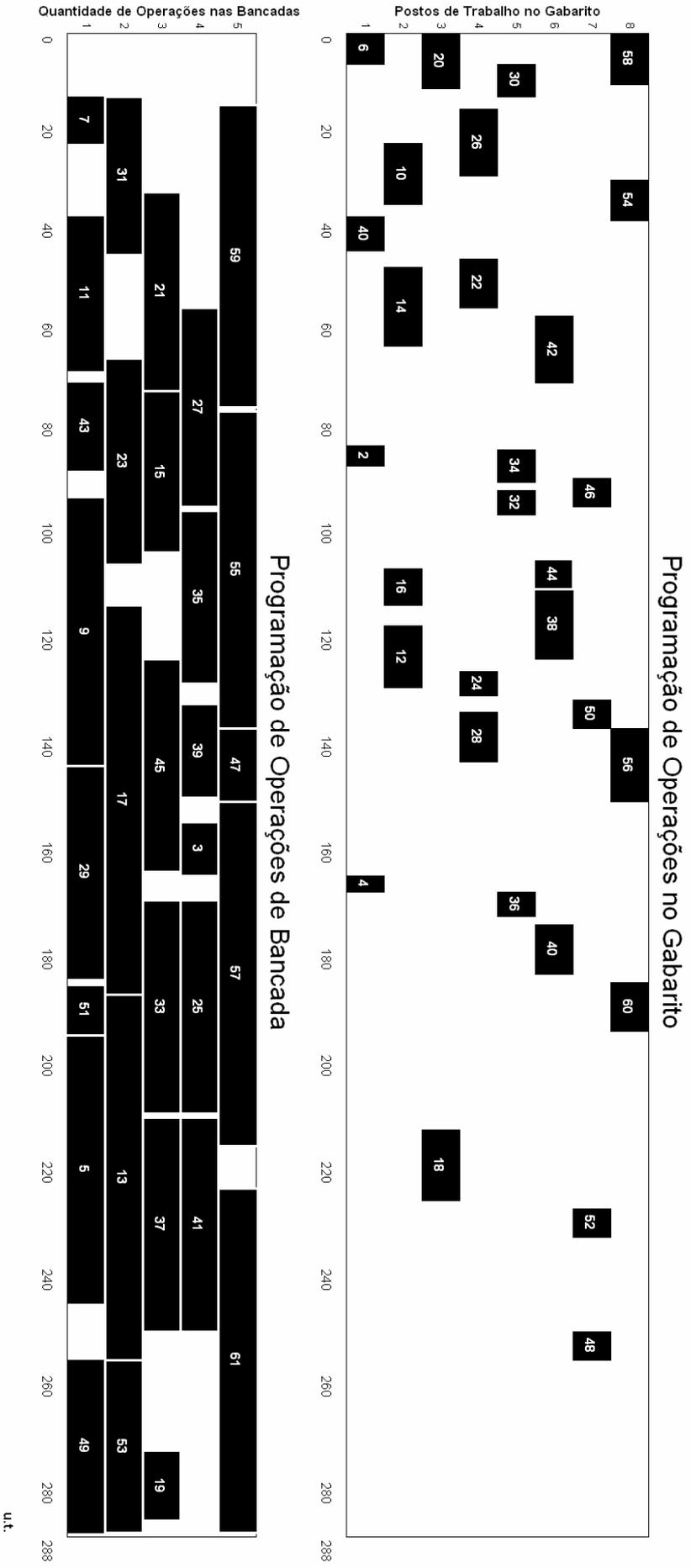


Figura 55: Gráfico de Gantt com o programa de produção do exemplo 2 na fase 2

Como nesse caso a função objetivo é minimizar o uso de mão de obra, que é um recurso, o tempo total para completar as tarefas é utilizado quase no limite do tempo de projeto, que é de 288 u. t. Observando a figura 56, pode-se observar que o padrão do programa de produção no gabarito possui diversos momentos de ociosidade. A figura 56 apresenta a variação da quantidade de mão de obra trabalhando em cada uma das equipes de montagem, de gabarito e de bancada, ao longo do programa de produção.

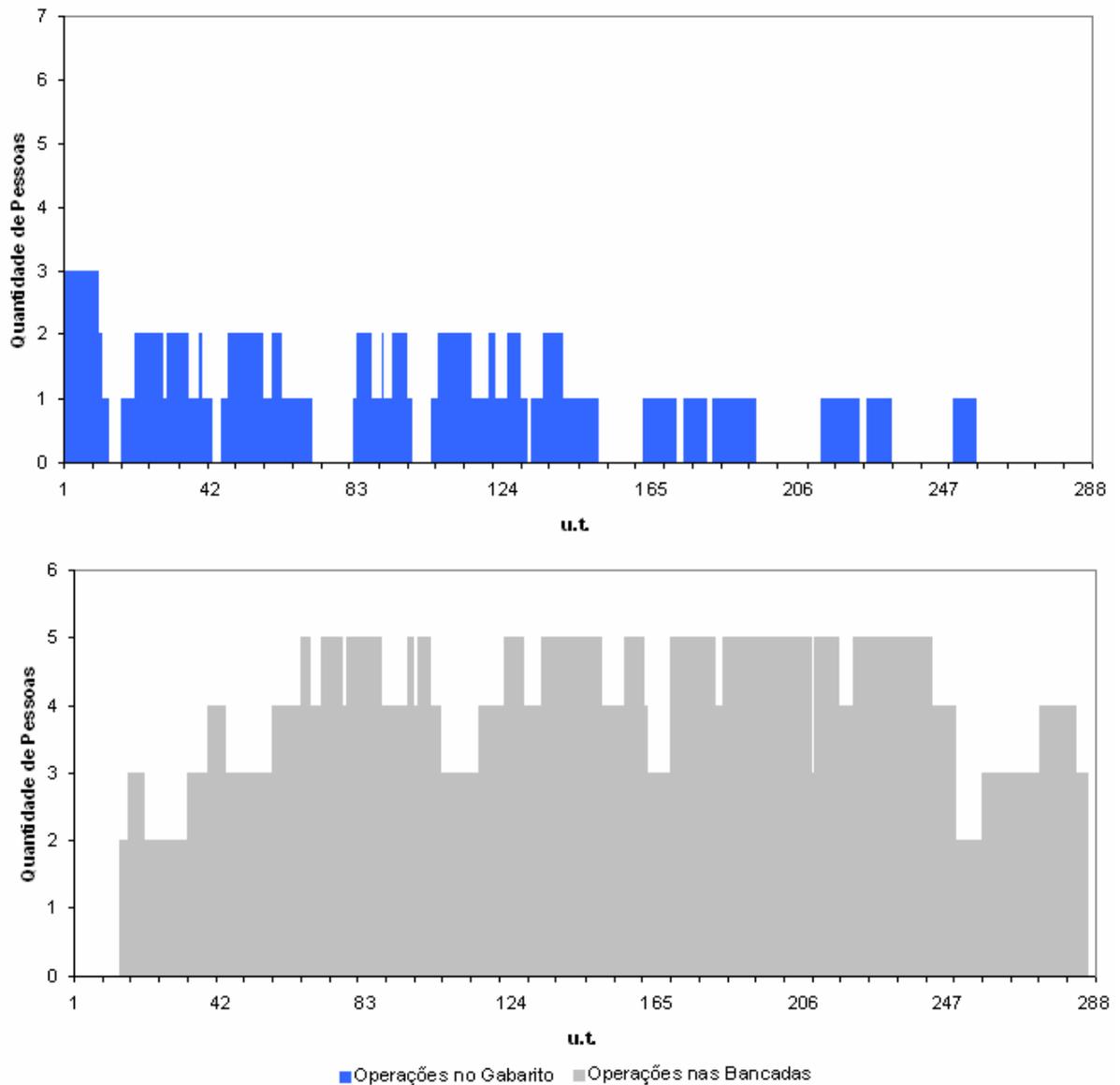


Figura 56: Variação do número de operadores no gabarito e na bancada para a fase 2 do subconjunto

A solução ótima indica o uso de 2 operadores de gabarito e 5 de bancada, ou seja, 7 operadores no total. A figura 57 apresenta a variação do número total de operadores trabalhando nas operações de montagem.

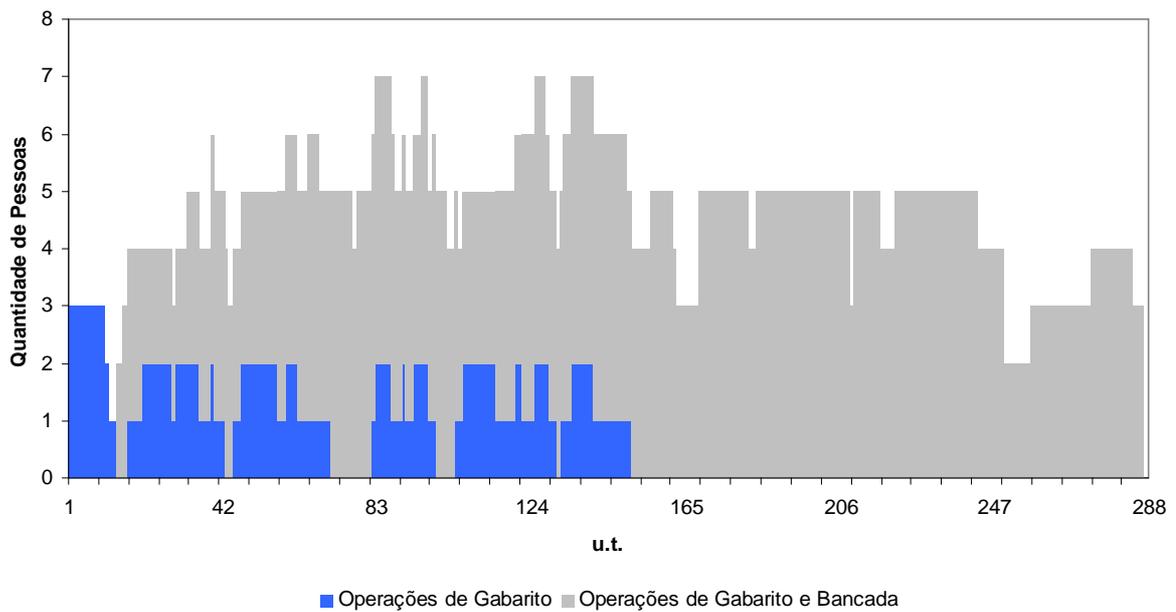


Figura 57: Variação do número de operadores na fase 2

A variação da quantidade de mão de obra na solução da figura 57 tem desvio padrão de 1,19, que é maior que da fase 1. Resolvendo o mesmo problema, porém, variando o *makespan* entre o mínimo possível, 161 u. t., e 288 u. t., pode-se determinar como o tamanho ótimo da equipe de montagem varia conforme o *makespan*, ou a cadência de produção, varia. A figura 58 apresenta essa variação para o modelo (52)-(68) na fase 2. Perceba que a quantidade de mão de obra apresenta um padrão “escada”, com degraus discretos. Para um dado tamanho de mão de obra, quanto menor o *makespan*, menor é a capacidade de a equipe responder a variações no sistema produtivo. No limite de mudança de tamanho para uma quantidade maior, uma pequena variação na produção precisa de mais uma pessoa para ser absorvida. No limite para uma quantidade menor, a equipe consegue absorver maiores mudanças sem precisar aumentar de tamanho, apenas mudando o ritmo de produção. A figura 58 e as outras curvas de mão de obra em função do *makespan* apresentadas nos resultados das outras fases ilustram uma forma rápida e fácil do tomador de decisão entender e decidir qual tamanho de equipe atenderia novos valores de cadência de produção.

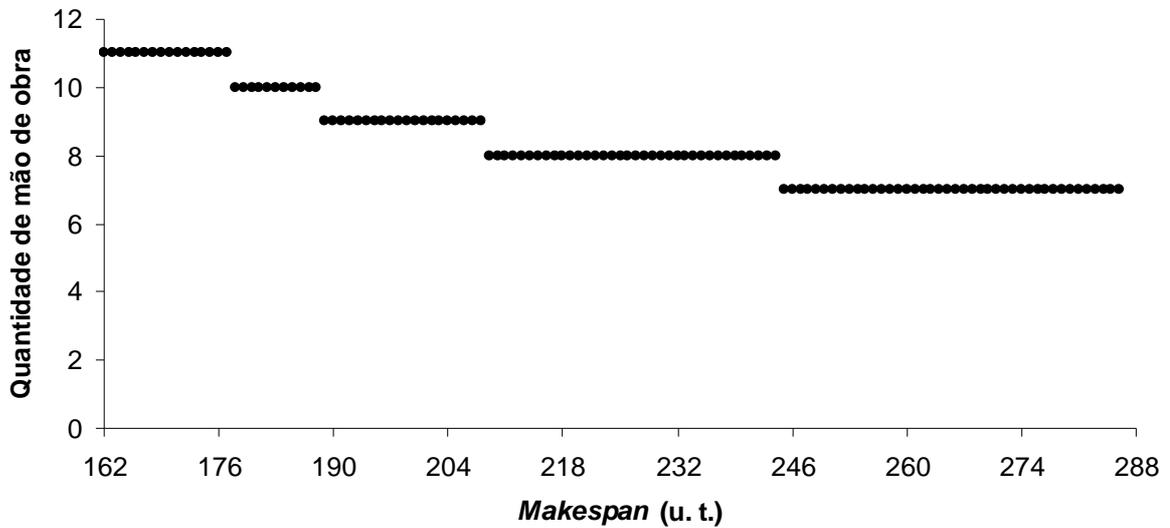


Figura 58: Tamanho ótimo das equipes de montagem em função do *makespan* para a fase 2

O tempo computacional para encontrar cada uma das soluções apresentadas na figura 58 é apresentado na figura 59. O tempo computacional máximo do CPLEX foi definido como 600 minutos. Assim, os pontos com tempo de 600 minutos na figura 59 significam que o CPLEX não encontrou a solução ótima ou não conseguiu provar que a solução encontrada era ótima, dentro do tempo estipulado. Apesar disso, é possível observar que, em uma grande quantidade de casos, o CPLEX encontra solução ótima em menos de 100 minutos. As soluções da figura 58 são duplamente restritas, ou seja, se para um dado *makespan* é conhecido o valor ótimo para o mesmo problema, mas com o *makespan* maior em 1 unidade e menor em 1 unidade, então, a solução ótima deve estar contida dentro das duas soluções ótimas conhecidas. Se as duas soluções conhecidas têm a mesma solução ótima, ou seja, a mesma quantidade de pessoas, logo, o projeto com *makespan* entre as duas soluções conhecidas já é conhecido, sem a necessidade de utilizar o CPLEX para descobrir a solução ótima.

Por exemplo, é possível observar que na figura 58 o *makespan* de 218 tem solução ótima de 8 pessoas. Suponha que o CPLEX encontrou a solução ótima para o *makespan* de 217 e 219 e ambas as soluções são de 8 pessoas. Assim, a solução do projeto com *makespan* 218 é 8 pessoas, pois, na pior hipótese, a solução com *makespan* 217 é uma solução factível para o *makespan* 218 e se a solução 219 é factível, logo, o problema com *makespan* 218 também é. Dessa forma, em alguns casos em que o CPLEX não consegue

provar otimalidade da solução, ainda assim é possível conhecer qual a solução ótima, dependendo dos valores e da otimalidade das soluções vizinhas.

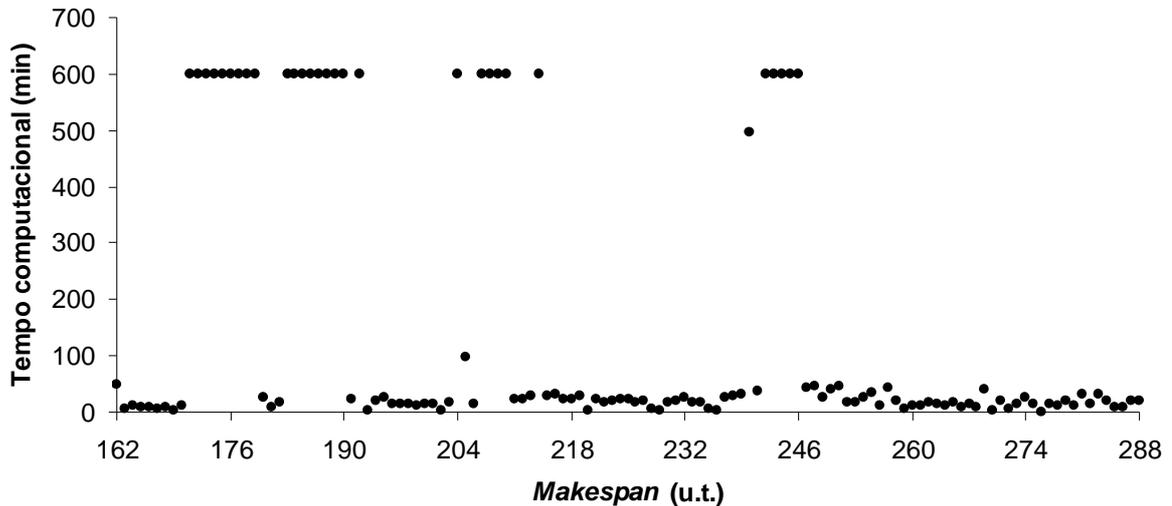


Figura 59: Tempo computacional para cada um dos *makespan* na fase 2

### 5.1.3 Exemplo computacional da fase 3

Nessa fase, dois modelos de programação de projetos podem ser utilizados, conforme apresentado na seção 4.4. O modelo (59)-(67), que é o modelo de programação de projetos com múltiplos modos de execução, tem 35.713 variáveis binárias, 17.542 equações e o CPLEX procurou a solução ótima por 600 minutos, mas não a encontrou. O modelo (68)-(75), que é modelo de programação de projetos com único modo, mas utilizando a variável de folga  $y_t$ , tem 18.145 variáveis inteiras, 8.349 equações e o CPLEX demorou 29 minutos para encontrar a solução ótima e provar a sua otimalidade. A solução encontrada pelo modelo (59)-(67) precisa de 6 pessoas, enquanto que a solução do modelo (68)-(75), que é ótima, precisa de 5 pessoas. Por isso, serão apresentados o programa de produção e a variação da quantidade de mão de obra do programa de produção apenas do modelo (68)-(75). A figura 60 ilustra o programa de produção da solução e a figura 61 apresenta a variação de mão de obra para as montagens no gabarito e nas bancadas do modelo (68)-(75). Assim como na fase 2, as operações de gabarito apresentam vários momentos de ociosidade ao longo do programa de produção.

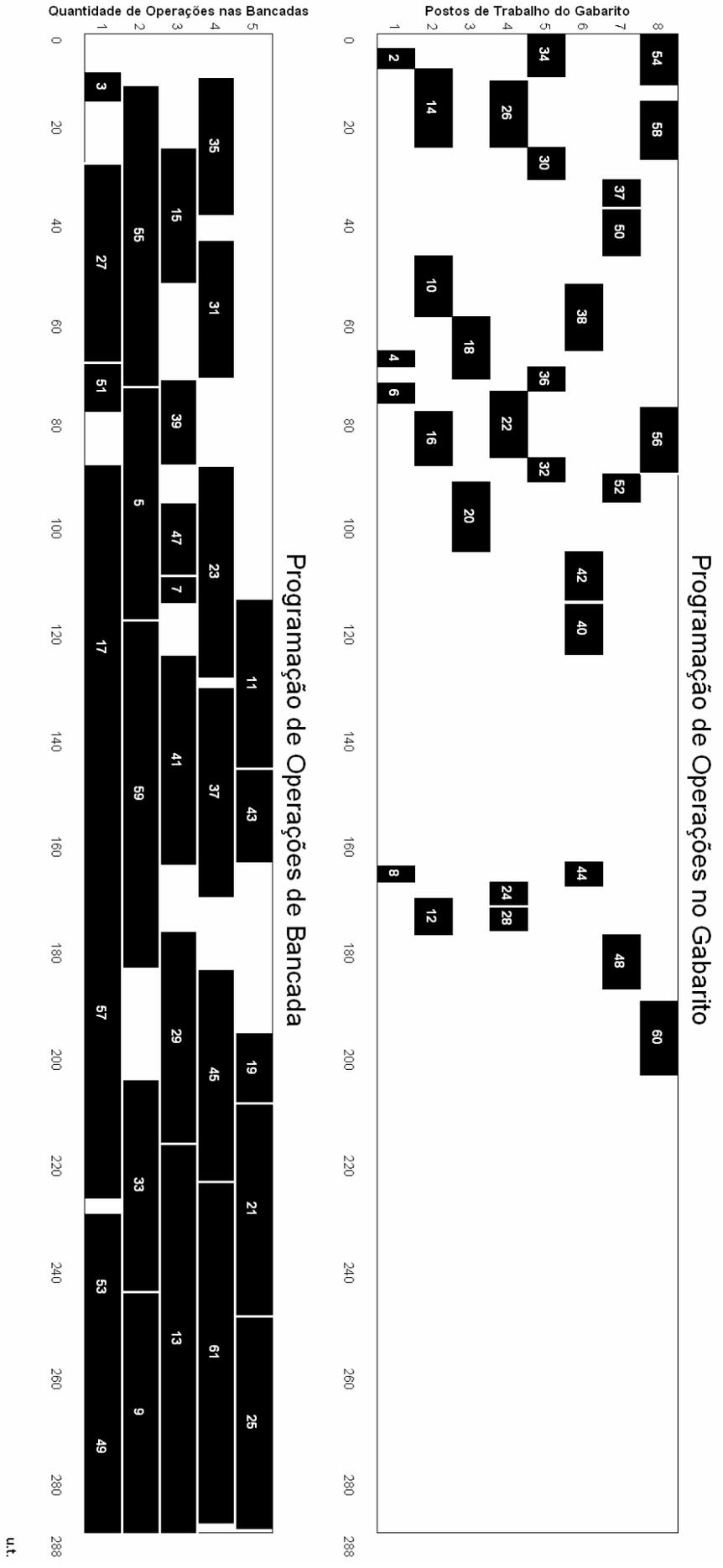


Figura 60: Gráfico de Ganti com o programa de produção do exemplo 2 na fase 3 do modelo (68)-(75)

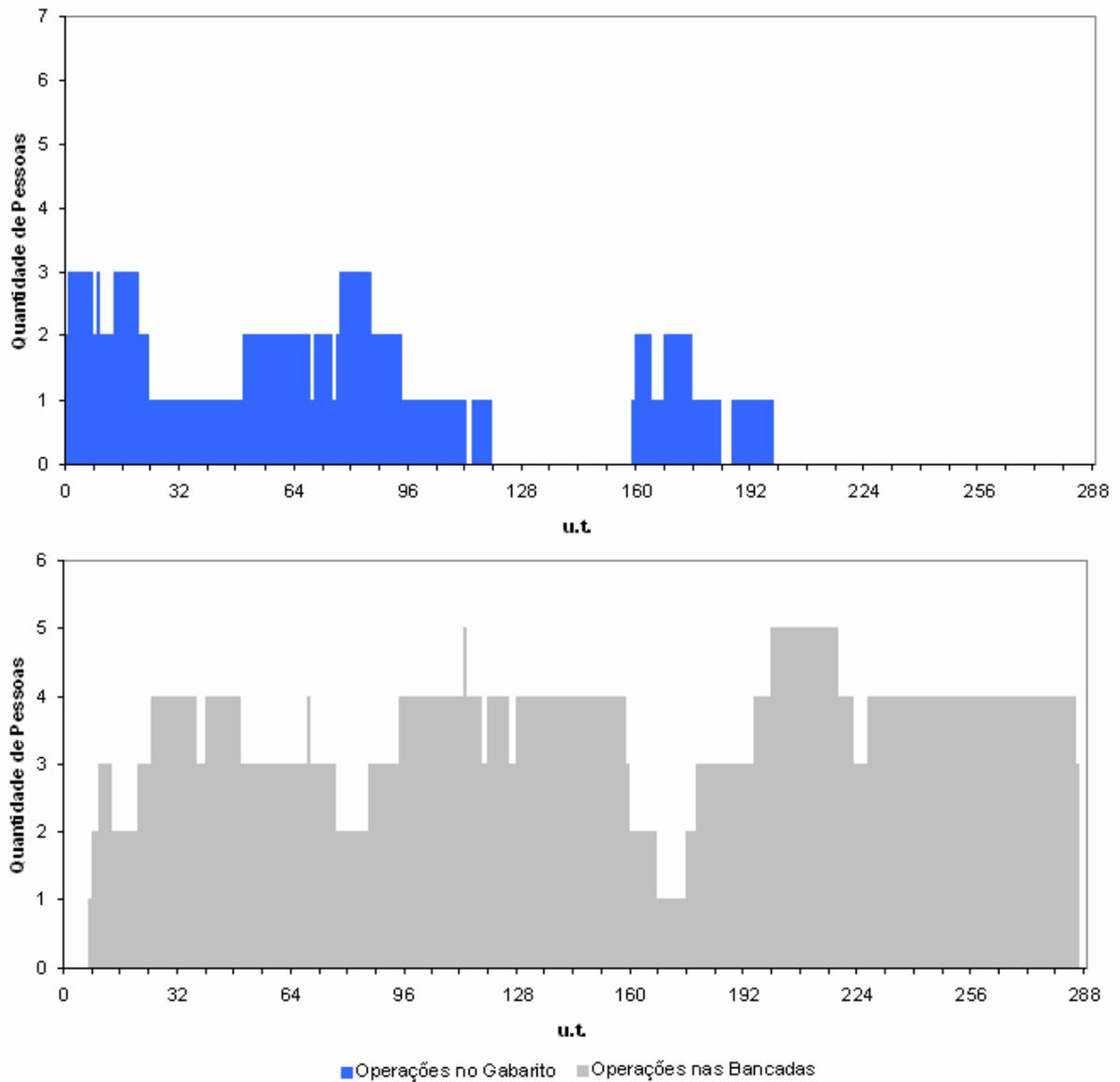


Figura 61: Variação do número de operadores no gabarito e na bancada para a fase 3 do modelo (68)-(75)

A figura 62 ilustra a quantidade total de operadores necessários ao longo do tempo para executar a montagem do avião. Note que são necessários cinco operadores, capazes de trabalhar nas bancadas e nos gabaritos, para terminar o programa, dois a menos que na fase 2.

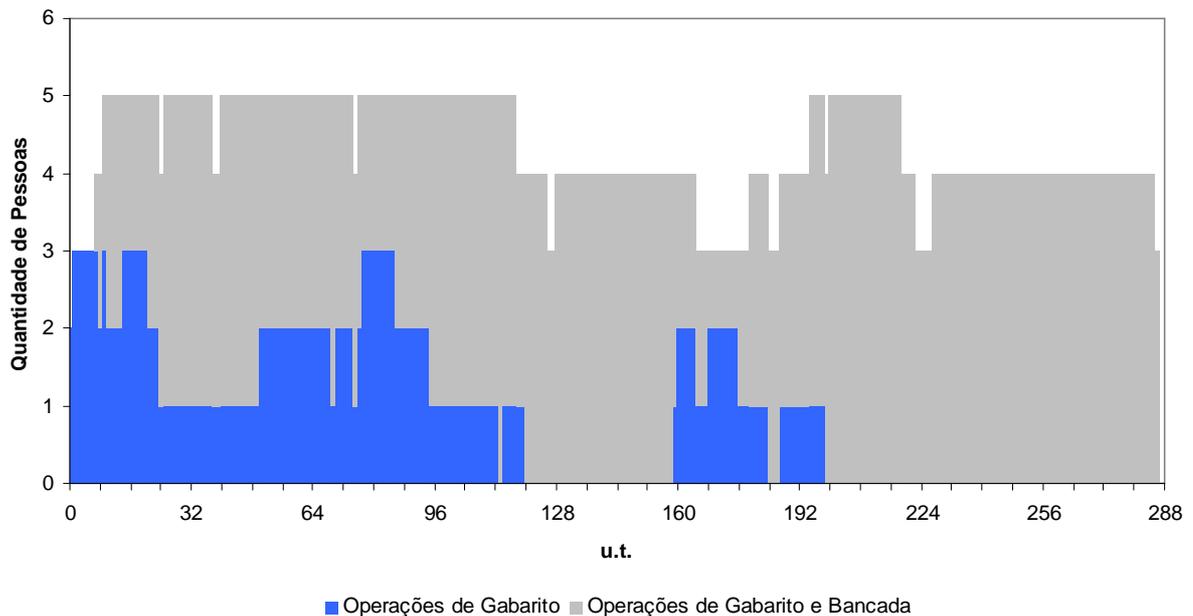


Figura 62: Variação do número de operadores no exemplo 2 na fase 3 do modelo (68)-(75)

A solução apresenta um nivelamento da mão de obra melhor que a fase 1 e 2, com média de 4,3 pessoas e desvio padrão de 0,67 pessoas. Na solução ótima, a quantidade de operadores flexíveis sugerida foi cinco, ou seja, todos os operadores seriam flexíveis. Caso não exista disponível toda essa quantidade de mão de obra para gabarito, uma restrição adicional poderia ser introduzida no modelo apresentado na seção 4.3, limitando a variável ao valor disponível.

Na figura 63 e na figura 64 são apresentados respectivamente os tamanhos ótimos da equipe de montagem, conforme o *makespan* para os modelos (59)-(67), de programação de projetos com múltiplos modos de execução, e o modelo (68)-(75), de programação de projetos utilizando a variável de folga. O modelo (59)-(67) encontra algumas soluções piores que o modelo (59)-(67) porque em várias instâncias o tempo máximo de processamento de 600 minutos é atingido antes que a solução ótima seja encontrada. O perfil é o mesmo da figura 58, com vários “degraus”. As mesmas observações da figura 58 valem para a figura 64.

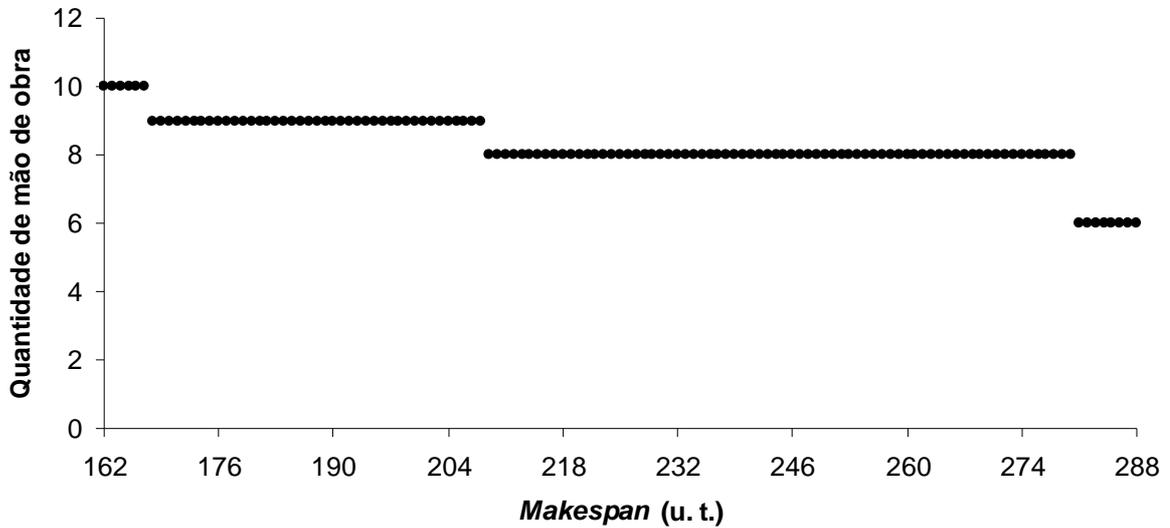


Figura 63: Tamanho das equipes de montagem em função do *makespan* para a fase 3 do modelo (59)-(67)

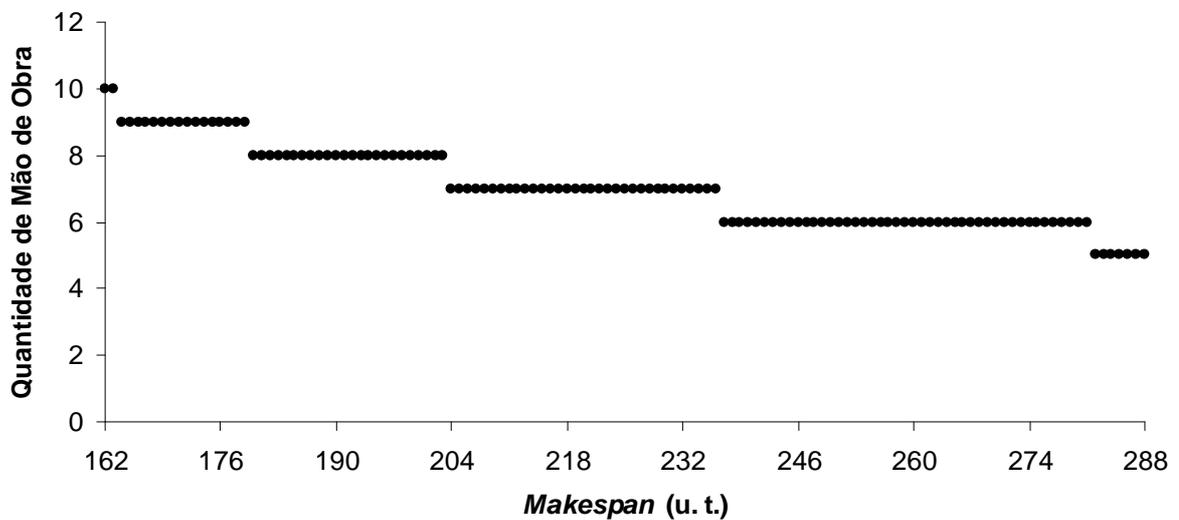


Figura 64: Tamanho ótimo das equipes de montagem em função do *makespan* para a fase 3 do modelo (68)-(75)

A figura 65 e a figura 66 apresentam o tempo computacional que o CPLEX demorou para encontrar cada uma das soluções dos modelos (59)-(67) e (68)-(75), respectivamente. Em alguns casos do modelo (59)-(67), o CPLEX não encontrou solução ótima dentro do tempo determinado de 600 minutos. O modelo (68)-(75), como era esperado, apresentou um desempenho computacional melhor que do modelo (59)-(67). O tempo médio computacional de todas as instâncias, inclusive das que não encontraram solução ótima dentro

do tempo de 600 minutos, do modelo (59)-(67) foi de 351 minutos e do modelo (68)-(75) foi de 61 minutos.

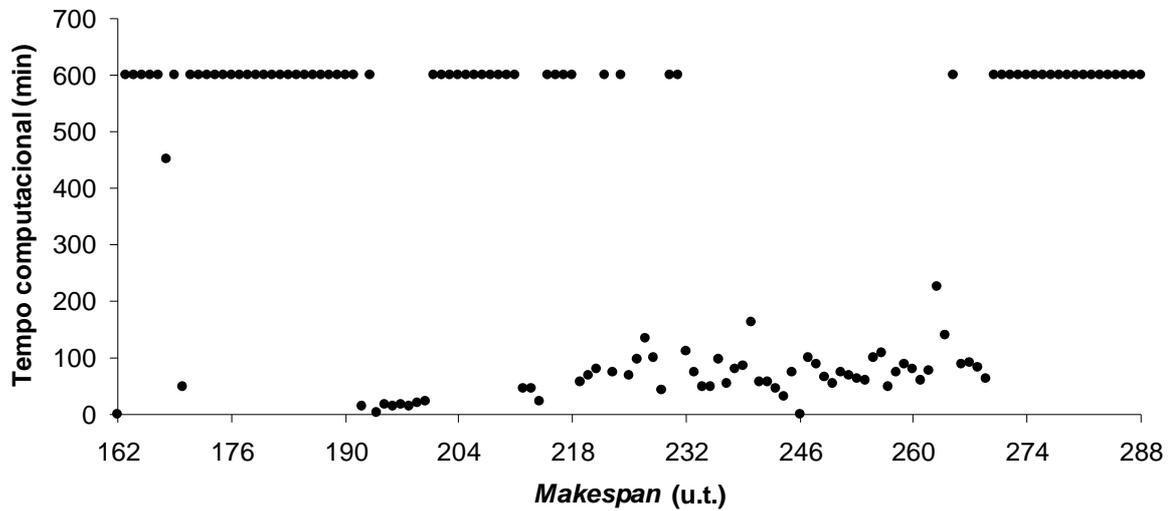


Figura 65: Tempo computacional para cada um dos *makespan* na fase 3 do modelo (59)-(67)

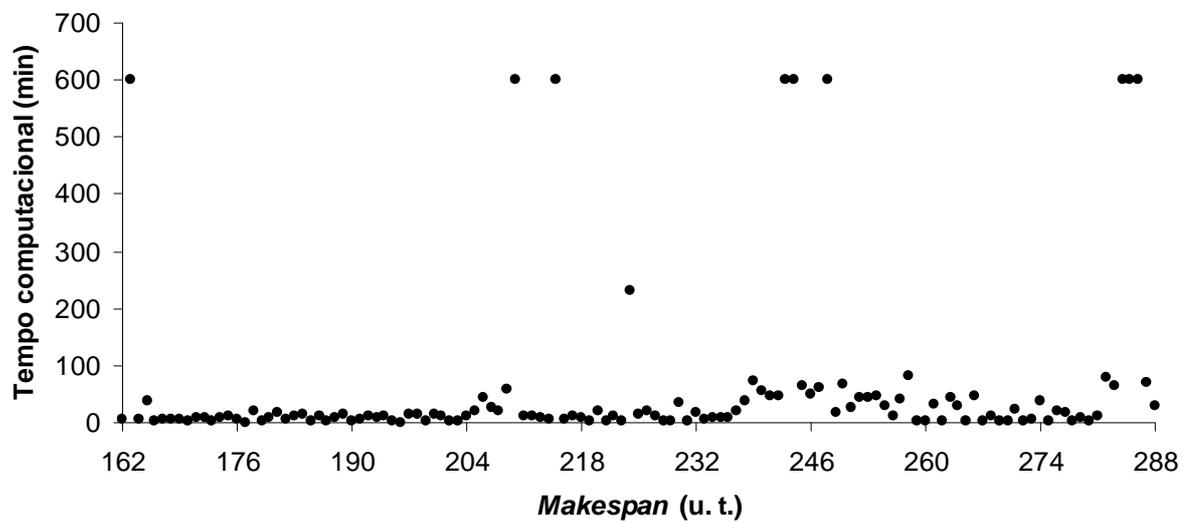


Figura 66: Tempo computacional para cada um dos *makespan* na fase 3 do modelo (68)-(75)

#### 5.1.4 Exemplo computacional da fase 4

Os dados utilizados nesse exemplo são os mesmos do exemplo 2, apresentados na tabela 6 no apêndice A. O modelo utilizado nessa fase é o modelo (52)-(58), que possui 17.856 variáveis e 8.061 equações. O CPLEX demorou 49 minutos para encontrar a solução ótima. A figura 67 apresenta o programa de produção da solução encontrada.

Na fase 4, existe apenas um único tipo de mão de obra, que é capaz de realizar qualquer operação de montagem. Por isso, é esperado que o recurso mão de obra seja melhor utilizado na fase 4 comparada as outras fases, depois de nivelado, devido à flexibilidade do recurso mão de obra. A solução encontrada na fase 3 é uma solução viável para a fase 4, caso a rede de precedência e as durações das tarefas sejam as mesmas. Desta forma, na pior das hipóteses, a solução da fase 4 deve ser igual ou melhor que a solução da fase 3, considerando o objetivo de minimizar a quantidade de mão de obra.

A figura 68 apresenta a variação da quantidade de mão de obra nas operações de gabarito e de bancada. A figura 69 ilustra a variação da quantidade total de mão de obra para todo o programa de produção. A média da quantidade de mão de obra é de 4,77 e a seu desvio padrão é de 0,59 pessoas.

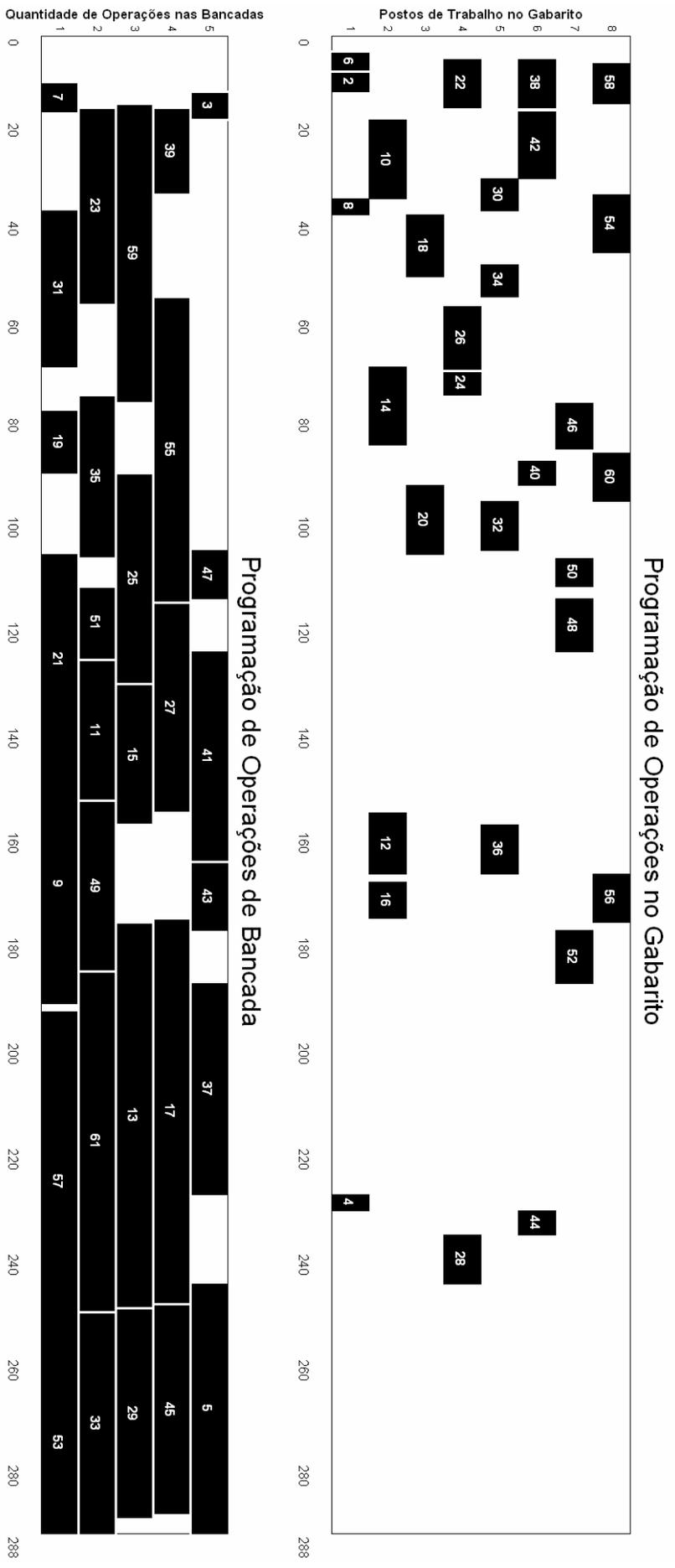


Figura 67: Gráfico de Gantt com o programa de produção do exemplo 2 na fase 4 do modelo (52)-(58)

u.t.

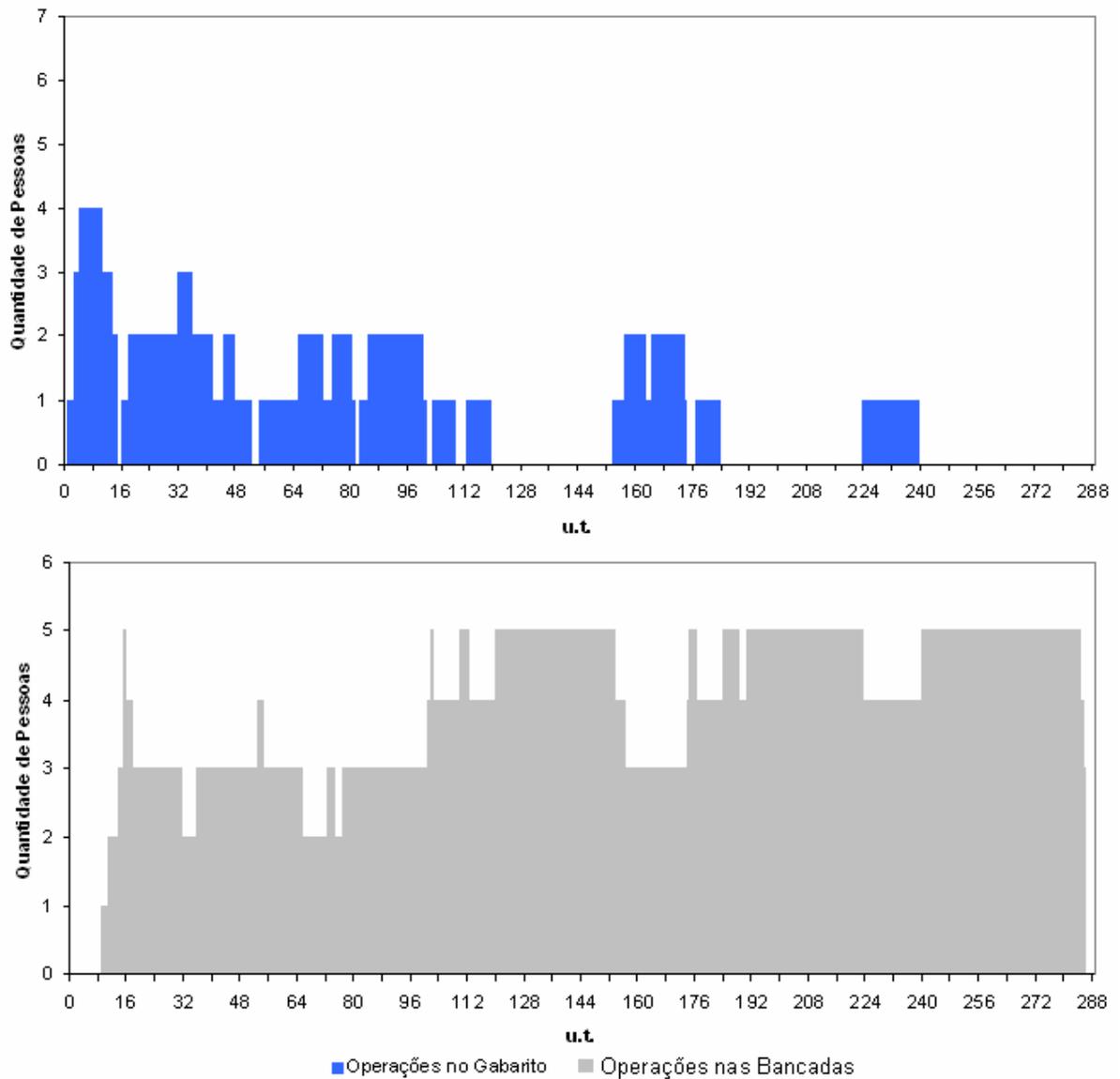


Figura 68: Variação do número de operadores no gabarito e na bancada para a fase 4 do modelo (52)-(58)

Note que na solução apresentada são necessários 5 operadores no total, porém, comparando-se a figura 69 com a figura 62, é possível perceber que na fase 4 serão necessários menos operadores que na fase três em vários momentos do programa de produção.

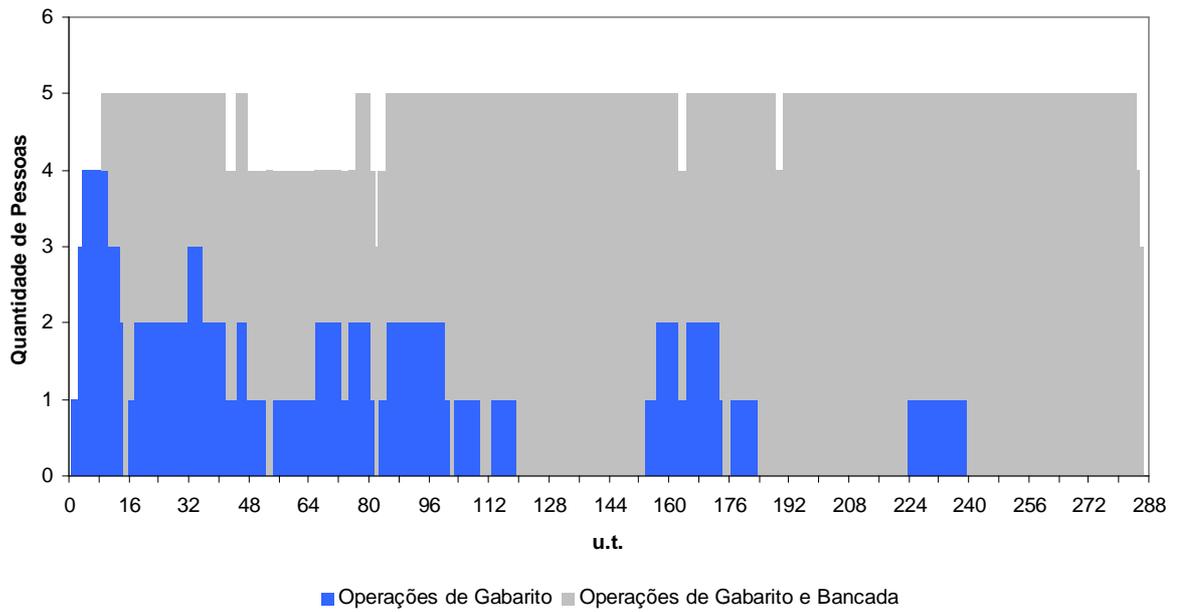


Figura 69: Variação do número de operadores no exemplo 2 na fase 4 do modelo (52)-(58)

A figura 70 apresenta a variação do tamanho da equipe conforme o tempo de ciclo desejado. Nesse caso, a equipe varia entre 10 pessoas até 5, conforme a variação do *makespan*.

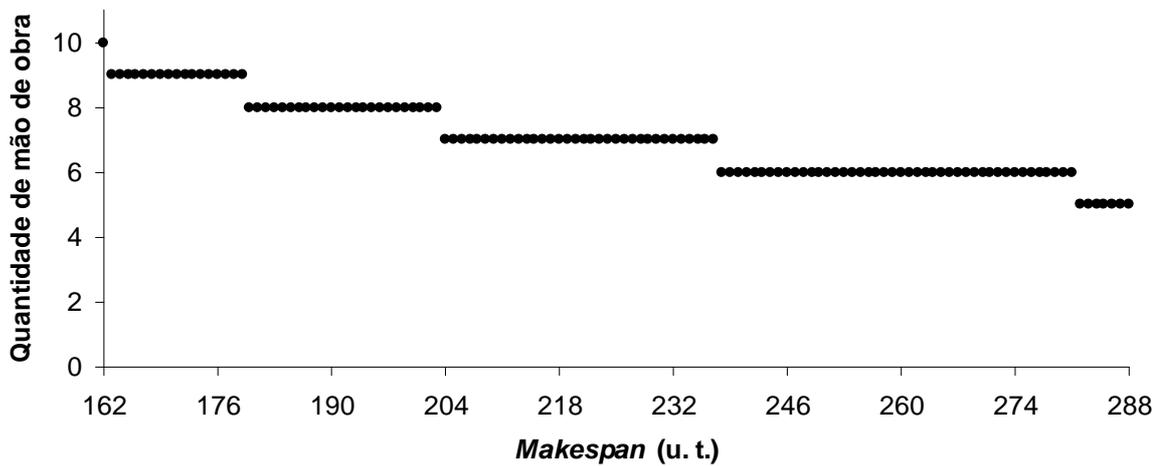


Figura 70: Tamanho ótimo das equipes de montagem em função do *makespan* do modelo (52)-(58)

A figura 71 apresenta o tempo computacional que o CPLEX utilizou para encontrar a solução ótima. Nesse caso, praticamente todas as instâncias tiveram tempo computacional inferior a 100 minutos. Comparando a figura 71 com a figura 66 e a figura 59,

é possível perceber que a fase 4 precisa de um menor tempo computacional para encontrar as soluções ótimas.

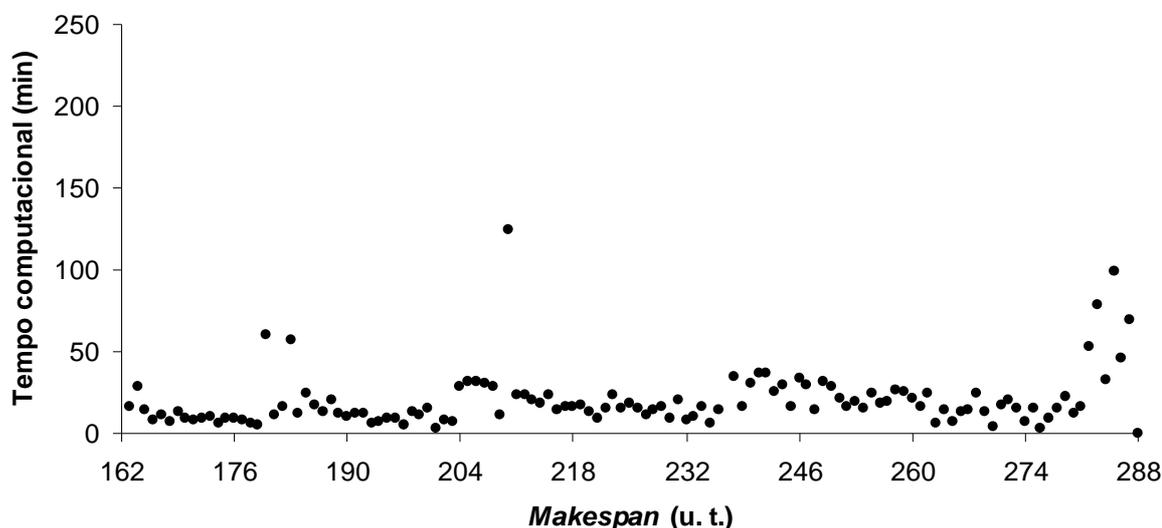


Figura 71: Tempo computacional para cada um dos *makespan* na fase 4 do modelo (52)-(58)

A tabela 3 resume os tempos computacionais dos modelos e as quantidades de variáveis de cada um. O modelo fase 3, com múltiplos modos de execução, exige maior esforço computacional por ser o modelo com maior quantidade de variáveis binárias, aproximadamente o dobro dos modelos da fase 2 e 4, pois se trata do mesmo modelo de programação de projetos, mas na versão com múltiplos modos de execução.

Tabela 3: Comportamento computacional dos modelos

	<b>Tempo computacional (min)</b>	<b>Variáveis Inteiras</b>	<b>Variáveis contínuas</b>
Fase 1 exemplo 1	0,017	464	436
Fase 1 exemplo 1	154	10.046	0
Fase 1 exemplo 2	0,017	278	310
Fase 1 exemplo 2	8	14.506	0
Fase 2	49	17.866	0
Fase 3 múltiplos modos	600	35.713	0
Fase 3	29	18.145	0
Fase 4	49	17.856	0

A tabela 4 apresenta o conjunto dos resultados encontrados nas soluções ótimas dos modelos apresentados. É possível perceber que, desde a fase 1, há uma significativa melhora na quantidade de mão de obra necessária para realizar cada montagem, reduzindo de 9, após o nivelamento da mão de obra na fase 1, para 5. Apesar de o *makespan* ser

consideravelmente diferente, ainda assim são resultados importantes do ponto de vista prático. A média foi calculada somando-se as quantidades de pessoas trabalhando ao mesmo tempo, e dividindo pela quantidade de períodos existentes para todo o horizonte de tempo do programa de produção. A variância é a dispersão dos valores ao redor da média calculada.

Tabela 4: Resultados das soluções ótimas

	<b>Média (pessoas)</b>	<b>Desvio Padrão (pessoas)</b>	<b>Número total de operadores</b>	<b>Makespan</b>
Fase 1 exemplo 1	8,9	3,28	14	195
Fase 1 exemplo 1 nivelado	8,9	2,02	10	195
Fase 1 exemplo 2	7,9	3,54	13	161
Fase 1 exemplo 2 nivelado	8,5	1,09	9	161
Fase 2	4,77	1,19	7	288
Fase 3	4,33	0,67	5	288
Fase 4	4,77	0,59	5	288

O modelo fase 3 apresenta uma média de operadores menor que o modelo fase 4. Isso se deve, à menor quantidade de operadores no início do programa. É importante ressaltar que em ambas as soluções, embora sejam programas de produção diferentes, a quantidade total de operadores é a mesma. Assim, a solução da fase 3 poderia ser uma solução alternativa da fase 4, sem mudar o valor da função objetivo.

## 5.2 Discussões finais

Todos os casos, das 4 fases, foram estudados considerando dados reais fornecidos pela empresa. As soluções ótimas encontradas são entre 10 e 20% melhores do que as obtidas na prática pela empresa. Assim, os modelos propostos conseguem melhorar significativamente o uso dos recursos produtivos, utilizando um tempo computacional aceitável para o contexto de decisão envolvida. A determinação do menor *makespan* na fase 1, por exemplo, gerou uma informação até então desconhecida na prática da empresa, que é a capacidade máxima de produção em cada gabarito. Com essa informação, é possível realizar o planejamento de duplicações de gabaritos de produção, que são recursos caros e que necessitam de vários meses para serem produzidos. Na fase 1, foram geradas instâncias em que são montados 2 aviões ao mesmo tempo e a solução mostrou que, caso exista a quantidade de mão de obra disponível, é possível realizar a montagem de 2 aviões com o mesmo *makespan* encontrado com a montagem de um único avião, tanto para o exemplo 1

como para o exemplo 2, sem a necessidade de duplicação de gabarito. Isso significa que a capacidade de produção pode ser maior que a apresentada nas soluções dos exemplos na fase 1, caso a empresa aceite realizar a montagem de 2 aviões ao mesmo tempo no mesmo gabarito.

As curvas de mão de obra em função do *makespan* também são informações de grande importância na prática. Variações de cadência de produção podem ser avaliadas rapidamente apenas observando qual seria o novo *makespan* e a respectivo tamanho ótimo da equipe, para qualquer uma das fases de produção. Também é possível avaliar melhor qual a quantidade de variação na cadência de produção que um determinado tamanho de equipe suporta, sem prejudicar a entrega dos subconjuntos no prazo estipulado. O tomador de decisão pode, por exemplo, escolher ter uma equipe com mais pessoas para atender variações aleatórias do processo produtivo, ao invés de ficar em um limite do tamanho de equipe em que a capacidade de resposta da equipe às variações é pequena. As soluções sugerem que a utilização de uma estratégia para agilizar o aprendizado dos operadores e percorrer as fases de produção mais rapidamente pode gerar melhoria da produtividade. Na indústria aeronáutica, uma das discussões recorrentes sobre como organizar a produção é a utilização da estratégia de especializar os operadores em certas montagens, para ganhar qualidade, ou a estratégia de formar equipes polivalentes, para ganhar flexibilidade e produtividade. Essa é ainda uma questão aberta e, portanto, deve-se avaliar caso a caso, para decidir qual das duas estratégias apresenta menor custo total.

Por simplicidade, esse trabalho desconsidera qualquer variação na qualidade da produção ao percorrer as fases da curva de aprendizagem, ou seja, supõe-se que as equipes somente mudam de fase quando estão com um grau de maturidade suficiente para realizar a mudança, sem prejudicar a qualidade da produção.

## 6 Conclusões e perspectivas

O objetivo desse trabalho foi estudar a programação de montagens em um tipo específico de gabarito presente na indústria aeronáutica, os gabaritos múltiplos com restrições de adjacência. A crescente competição entre as empresas de construção de aviões, especialmente no seguimento de aeronaves comerciais com até 120 passageiros, fazem com que várias oportunidades de pesquisa, desenvolvimento e inovação nos métodos e processos de produção sejam buscadas pelas empresas para reduzir custos e assim, surgem grandes oportunidades para as técnicas de pesquisa operacional.

A programação da produção aplicada às montagens em gabaritos da indústria aeronáutica, mais especificamente no contexto dos gabaritos com restrições de adjacência estudados neste trabalho, mostrou que existem potenciais de ganhos de produtividade a serem explorados pelas empresas deste setor. Modelos matemáticos inspirados em problemas de *job shop* e programação de projetos foram capazes de representar adequadamente o problema e encontrar soluções ótimas na maior parte das vezes, em tempos aceitáveis para o contexto de tomada de decisão do problema. As soluções práticas na empresa são construídas por *softwares* que geram soluções visuais, em gráficos de Gantt, que então são niveladas manualmente. Os modelos matemáticos conseguem encontrar soluções ótimas em tempos da ordem de cerca de 5 minutos na maior parte dos casos, enquanto as soluções manuais demoram várias dezenas de minutos, sem garantia de otimalidade. Assim, a utilização dos modelos propostos diminui o tempo necessário pelos analistas de planejamento e programação da produção para elaborar um novo programa de produção.

A partir de um estudo de caso realizado em uma empresa aeronáutica, as soluções encontradas são entre 10 e 20% melhores do que as soluções utilizadas na prática. Além disso, Estas soluções também auxiliam na determinação da capacidade máxima de cada gabarito de montagem, e também na análise de como o tamanho ótimo das equipes varia em função da cadência de produção, informações até então pouco exploradas na prática da empresa estudada. Isso permite ao responsável pela montagem planejar melhor o tamanho da equipe e possíveis necessidades de treinamento, conforme previsões de alteração na cadência de produção dos aviões.

Uma perspectiva interessante de trabalho futuro seria estudar a programação de montagens em gabaritos com durações incertas, utilizando técnicas de programação estocástica e otimização robusta. Outra perspectiva importante seria estudar outras montagens que utilizam gabaritos na indústria aeronáutica ou em outros contextos (por exemplo, na

indústria naval, especialmente na produção de embarcações leves) e assim, verificar se os modelos apresentados nesse trabalho também poderiam representar adequadamente esses outros problemas. Nesse trabalho foi estudada uma montagem que utiliza apenas um único gabarito, mas uma perspectiva de pesquisa seria estudar modelos de programação de projetos em montagens que utilizam vários gabaritos, seqüenciais ou em paralelo, explorando a alocação de mão de obra em cada gabarito, ou seja, utilizando modelos de programação de projetos com múltiplos modos de execução. Uma outra perspectiva seria o desenvolvimento de modelos de programação inspirados em modelos da teoria de *job shop* de forma a incluir as restrições de mão de obra. Como o tempo nesses modelos é uma variável contínua, o menor número de variáveis binárias poderia refletir em tempos computacionais menores do que os modelos inspirados em programação de projetos. Por último, outra linha de pesquisa a ser investigada seria o desenvolvimento de métodos heurísticos para resolver os modelos das fases 1-4, visando reduzir os tempos computacionais e gerar soluções comparáveis em qualidade às dos métodos exatos.

Experimentos adicionais podem ser feitos estudando os efeitos da diferenciação de custo da mão de obra de gabarito e de bancada. Outra perspectiva é utilizar os modelos de cada fase em um contexto de curva de aprendizagem, considerando a redução de horas homem necessárias para montar os subconjuntos de um avião. Particularmente, estudar os efeitos da curva de aprendizagem no modelo matemático da fase 1 seria uma interessante forma de determinar como a redução dos tempos de montagem afetaria a capacidade de produção máxima do gabarito.

## 7 Referências

ABUABARA, A. **Otimização no corte de tubos estruturais**: aplicação na indústria aeronáutica agrícola. 2006. 190 f., Dissertação (Mestrado em Engenharia de Produção) – Universidade Federal de São Carlos, São Carlos, 2006.

ABUABARA, A., MORABITO, R. Modelos de programação inteira mista para o planejamento do corte unidimensional de tubos metálicos na indústria aeronáutica agrícola. **Gestão e Produção**, v. 15, n.3, pág. 605-617, set.-dez., 2008.

ABUABARA, A., MORABITO, R. Cutting optimization of structural tubes to build agricultural light aircrafts. **Annals of Operations Research**, vol. 169, n. 1, pág. 149-165, jul., 2009.

**AERO VODOCHODY**. Disponível em: <<http://www.aero.cz/en/>> Acesso em 22 dez. 2009.

AIAB – Associação das Indústrias Aeroespaciais do Brasil. Disponível em:<<http://www.aiab.org.br>>. Acesso em 24 mar. 2010.

ANZANELLO, M. J., FOGLIATTO, F. S. Curvas de aprendizado: estado da arte e perspectivas de pesquisa. **Gestão e Produção**, v. 14, n. 1, pág. 109-123, jan. abr., 2007.

ARENALES, M. N., ARMENTANO, V. A., MORABITO, R., YANASSE, H. H. **Pesquisa operacional**: modelagem e algoritmos. Rio de Janeiro: Elsevier, 2006.

BARBOSA, G. F. **Aplicação da metodologia DFMA** – design for manufacturing and assembly no projeto e fabricação de aeronaves. 2007, 165 f. Dissertação de Mestrado (Mestrado em Engenharia Mecânica), Escola de Engenharia de São Carlos, Departamento de Engenharia Mecânica, São Carlos, 2007.

BAKER, K. **Introduction to sequencing and scheduling**. New York: John Wiley, 1974.

BAKER, B. S., COFFMAN Jr, G. Mutual exclusion scheduling. **Theoretical Computer Science**, vol. 162, pág. 225-243, 1996.

BERTAZZO, R. P. **A crise da indústria aeronáutica brasileira**: 1945-1968. 2003, 50 f. Monografia (Monografia em História). Instituto de Ciências Humanas e Letras, Universidade Federal de Juiz de Fora, Juiz de Fora, 2003.

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.

BLAZEWICZ, J., ECKER, K. H., PESCH, E., SCHMIDT, G., WEGLARZ, J. **Scheduling computer and manufacturing processes**. Berlin: Springer, 1996.

BOEING. Disponível em: [www.boeing.com](http://www.boeing.com). Acesso em: 22 dez. 2009.

BOMBARDIER AEROSPACE **Fuselage test barrel for Bombardier Cseries aircraft arrives**. 2009. Disponível em: <http://www.bombardier.com/en/aerospace/media-centre/press-releases/details?docID=0901260d800b632d>>. Acesso em: 21 dez. 2009.

BOTELHO, A. J. J. Da utopia tecnológica aos desafios da política científica e tecnológica: o Instituto Tecnológico de Aeronáutica (1947-1967). **Revista Brasileira de Ciências Sociais**, vol. 14, n. 39, pág. 139, 1999.

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. 2009a.

BRASIL. Ministério da Ciência e Tecnologia. Financiadora de Estudos e Projetos FINEP. Robôs garantem precisão na montagem de aeronaves. **Revista Inovação em Pauta**, vol. 8, pág. 27, nov./dez., 2009b.

BRÖTJE. Disponível em: < <http://www.claas-fertigungstechnik.com>>. Acesso em: 22 dez. 2009.

BRUCKER, P., DREXL, A., MÖHRING, R., NEUMANN, K., PESCH, E. Resource-constrained project scheduling: notation, classification, models, and methods. **European Journal of Operational Research**, vol. 112, pág. 3-41, 1999.

BRUCKER, P., KRÄMER, A. Polynomial algorithms for resource constrained and multiprocessor task scheduling problems. **Journal of Operational Research**, vol. 90, pág. 214-226, 1996.

BRYMAN, A. **Research methods and organization studies**. London: Unwin Hyman, 1989.

BUCKINGHAM, R., CHINATRAKARAN, V., CONKIE, R., FERGUSON, G., GRAHAM, A., LAZELL, A., LICHON, M., PARRY, N., POLLARD, F., KAYANI, A., REDMAN, M., SUMMERS, M., GREEN, B. Snake-arm robots: a new approach to aircraft assembly. **Society of Automotive Engineers Technical Papers**. 2006 Disponível em: <<http://www.sae.org/technical/papers/2006-01-3141>>. Acesso em: 24 jul 2009.

CARLIER, J., MOUKRIM, A., XU, H. The project scheduling problem with production and consumption of resources: a list scheduling based algorithm. **Discrete Applied Mathematics**, vol. 157, pág. 3631-3642, 2009.

CHIKONG, H., CHANG, H-M., LIN, C-H. Simulation and analysis for layout of aerospace sheet metal process with lean concept. **Materials Science Forum**, vol. 505-507, pág. 907-912, 2006.

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.

CROWSON, R. **Assembly processes: finishing, packaging and automation**. 2 ed., New York: CRC/Taylor & Francis Group, 2006.

DALE, R. Offline programming and simulation help Boeing use giant automated riveter on C-17 aircraft. **Industrial Robot: An International Journal**, vol. 28, n. 6, pág. 478-482, 2001.

DeVIEGLI, R., FEIKERT, E. One-up assembly with robots. **Society of Automotive Engineers Technical Papers**, 2008. Disponível em: <<http://www.electroimpact.com/company.asp?page=Publications>>. Acesso em: 27 de jul de 2009.

DROZDOWSKI, M. On the complexity of multiprocessor task scheduling. **Bulletin of the Polish Academy of Sciences Technical Sciences**, vol. 43, n. 3, 1995.

DROZDOWSKI, M. Scheduling multiprocessor tasks – and overview. **European Journal of Operational Research**, vol. 94, 215-230, 1996.

DUIN, C. W., SLUIS, V. D. On the complexity of adjacent resource scheduling. **Journal of Scheduling**, vol. 9, pág. 49-62, 2006.

DUMONT, A. S., **O que eu vi o que nós veremos**. Biblioteca Virtual de Literatura, 1918. Disponível em: <http://www.dominiopublico.gov.br>. Acesso em: 2 mar. 2010.

DYCKHOFF, H. A typology of cutting and packing problems. **European Journal of Operational Research**, vol. 44, pág. 145-159, 1990.

EKICI, O., YONGACOGLU, A. IEEE 802.11a throughput performance with hidden nodes. **IEEE Communications Letters**, vol. 12, n. 6, jun., 2008.

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](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](http://www.embraer.com.br/institucional/download/1_062-Prd-VAE-L450-500_Advance-P-09.pdf)>, 2009. Acesso em 20 jul. 2009.

EMBRAER **Embraer market outlook**. 2009-2028. Disponível em:<[www.embraer.com/outlook](http://www.embraer.com/outlook)>. Acesso em: 15 mai. 2010.

EPSTEIN, L., LEVIN, A. On bin packing with conflicts. **Lectures Notes in Computer Science**, vol. 4368, pág. 160-173, 2007.

EPSTEIN, L., LEVIN, A., STEE, R.V. Two dimensional packing with conflicts, **Acta Informatica**, vol. 45, n. 3, pág. 155-175, 2008.

EVOLUTION MAGAZINE **Lift-off for China's regional jet**. set., 2006. Disponível em: <<http://evolution.skf.com/zino.aspx?articleID=14989>>. Acesso em: 29 dez. 2009.

EVEN, G., SHANAR, S. Scheduling of a smart antenna: capacitated coloring of unit circular-arc graphs. **Lectures Notes in Computer Science**, vol. 4235, pág. 58-71, 2006.

EVEN, G., HALLDÓSSON, M. M., KAPLAN, L., RON, D. Scheduling with conflicts: online and offline algorithms. **Journal of Scheduling**, vol. 12, pág. 199-224, 2009.

FAYAWERRA, N., WEBB, P. Automated assembly of fuselage skin panels. **Assembly Automation**, vol. 27, n. 4, pág 234-355, 2007.

FERREIRA, M. J. B., ARAÚJO, R. D., MELLO, C. H., MARQUES, R. Relatório de acompanhamento setorial: indústria aeronáutica. Agência Brasileira de Desenvolvimento Industrial, Unicamp, 2009. Disponível em: <<http://www.abdi.com.br/?q=system/files/Aeron%C3%A1utica+IV+dez+09.pdf>>. Acesso em: 10 mar 2010.

FILHO, M. R., YOSHISAKI, H. T. Y. **Análise de estoques em uma cadeia logística de suprimento na indústria aeronáutica utilizando um modelo de dinâmica de sistemas.** In: Encontro Nacional de Engenharia de Produção, XXVI, Fortaleza, out. 2006. Disponível em: <[http://www.abepro.org.br/biblioteca/ENEGEP2006\\_TR450302\\_8108.pdf](http://www.abepro.org.br/biblioteca/ENEGEP2006_TR450302_8108.pdf)>. Acesso em: 22 mar 2010.

FORJAZ, M. C. S. **As origens da Embraer.** Tempo Social, v. 17, n. 1, pág. 281-298, 2004.

FREIXO, O. M. **Incorporação da gestão dos custos do ciclo de vida ao processo de desenvolvimento do produto da EMBRAER.** 2004, 188 f. Tese de Doutorado (Doutorado em Engenharia de Produção), Universidade Federal de São Carlos, Departamento de Engenharia de Produção, São Carlos, 2004.

GANDHAM, S., DAWANDE, M., PRAKASH, R. Link scheduling in wireless sensor networks: distributed edge-coloring revisited. **Journal of Parallel and Distributed Computing**, vol. 68, pág. 1122-1134, 2008.

GENDRAU, M., LAPORTE, G. SEMET, F. Heuristics and lower bounds for the bin packing problem with conflicts. **Computers & Operations Research**, vol. 31, pág. 347-358, 2004.

GHABOOSI, K., LATVA-AHO, M., POMALAZA-RÁEZ, C. A novel MAC protocol and layer two transmission scheduling algorithm for WLANs. **Telecommunication Systems**, vol. 37, pág. 3-18, 2008.

GOLDBARG, M. G., LUNA, H P. L., **Otimização combinatória e programação linear: modelos e algoritmos.** 2 ed., São Paulo: Elsevier, 2005.

GÓMEZ, T., HERNANDÉZ, M., MOLINA, J., LÉON, M. A., ALDANA, E., CABALLERO, R. A multiobjective model for forest planning with adjacency constraints. **Annals of Operations Research**, fev., 2009.

GOYCOOLEA, M., MURRAY, A. T., BARAHONA, F., EPSTEIN, R., WEINTRAUB, A. Harvest scheduling subject to maximum area restriction: exploring exact approaches. **Operations Research**, vol. 53, n. 3, mai-jun, pág. 490-500, 2005.

GROOVER, M. P. **Fundamentals of modern manufacturing**: materials, processes and systems. 3 ed. Hoboken: John Wiley and Sons, 2007.

GULDEMOND, T. A., HURINK, J. L., PAULUS, J. J., SCHUTTEN, J. M. J. Time-constrained project scheduling. **Journal of Scheduling**, vol. 11, pág. 137-148, 2008.

HARTMANN, J., MEELER, C., MISHULL, A., SMITH, A. Automated wing panel assembly for the A-340-600. **Society of Automotive Engineers Technical Papers**, 2000. Disponível em: <<http://www.electroimpact.com/company.asp?page=Publications>>. Acesso em: 24 jul. 2009.

HAX, A. C., CANDEA, D. **Production and inventory management**. Prentice-Hall: New Jersey, 1984.

HEIKE, G., RAMULU, M., SORENSON, E., SHANAHAN, P., MOINZADEH, K. Mixed model assembly alternatives for low-volume manufacturing: the case of the aerospace industry. **International Journal of Production Economics**, vol. 72, pág. 103-120, 2001.

HERROELEN, W., LEUS, R. Project scheduling under uncertainty: survey and research potentials. **European Journal of Operational Research**, vol. 165, pág. 289-306, 2005.

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

INSEAD. Global Innovation Index 2009-2010. Disponível em: <<http://www.globalinnovationindex.org>>. Acesso em: 13 mar 2010.

IRANI, S., LEUNG, V. Scheduling With Conflicts and Applications to Traffic Signal Control. **ACM Symposium on Discrete Algorithms**, Atlanta, 1996.

IRANI, S., LEUNG, V. Scheduling with conflicts on bipartite and interval graphs. **Journal of Scheduling**, vol. 6, pág. 287-307, 2003.

JACKSON, J. R. An extension of Johnson's results on job lot scheduling. **Naval Research Logistics Quarterly**, vol. 3, pág. 201-203, 1956.

JANSEN, K. An approximation scheme for bin packing with conflicts. **Journal of Combinatorial Optimization**, vol. 3, n. 4, pág 363-377, 1999.

JANSEN, K. The mutual exclusion scheduling problem for permutation and comparability graphs. **Information and Computation**, vol. 180, pág. 71-81, 2003.

JIANG, L. B., LIEW, S. C. Hidden-node removal and its application in cellular WiFi networks. **IEEE Transaction on Vehicular Technology**, vol. 56, n. 5, set, 2007.

JOHNSON, S. M. Optimal two and three-stage production schedules with setup times included. **Naval Research Logistics Quarterly**, vol. 1, pág. 61-67, 1954.

KAPLAN, L. **Scheduling with conflicts**. 2007, 40 f. Dissertação (Mestrado em Engenharia Elétrica) Departament of Electrical Engineering – Systems, Faculty of Engineering, Tel-Aviv University, 2007.

KLEINROCK, L., TOBAGI, F. A. Packet switching in radio channels: part I – carrier sense multiple-access modes and their throughput-delay characteristics. **IEEE Transactions on Communications**, vol. 23, n. 12, dez., 1975a.

KLEINROCK, L., TOBAGI, F. A. Packet switching in radio channels: part II – the hidden terminal problem in carrier sense multiple-access and the busy-tone solution. **IEEE Transactions on Communications**, vol. 23, n. 12, dez., 1975b.

KOLISH, R., PADMAN, R. An integrated survey of deterministic project scheduling. **Omega The International Journal of Management Science**, vol. 29, pág. 249-272, 2001.

KRAJEWKI, L. J., RITZMAN, L. P. **Operations management: strategy and analysis**. 4 ed., Addison-Wesley Publishing Company, 1996.

LEUNG, J. Y-T. **Handbook of scheduling: algorithms, models and performance analysis**. Boca Raton: CRC Press, 2004.

LI, C., WANG, X. Cooperative multibeamforming in ad hoc networks. **EURASIP Journal on Advances in Signal Processing**, vol. 2008, 2008.

MAGALHÃES, M. S. **Configuração de uma rede de distribuição e de serviços na indústria aeronáutica**. 2008. 111 f. Dissertação (Mestrado em Engenharia de Produção) – Escola Politécnica, Universidade de São Paulo, São Paulo, 2008.

MECHAM, M., ANSELMO, J. C. Mitsubishi leads japanese aircraft. **Aviation Week**, 2008. Disponível em: <[http://www.aviationweek.com/aw/generic/story\\_generic.jsp?channel=awst&id=news/aw031708p1.xml](http://www.aviationweek.com/aw/generic/story_generic.jsp?channel=awst&id=news/aw031708p1.xml)>. Acesso em: 10 nov. 2009.

MEGSON, T. H. G. **Aircraft structures for engineering students**. 3 ed., Elsevier Butterworth-Heinemann, Oxford. pág. 211-232, 1999.

MORABITO, R., PUREZA, V. Modelagem e Simulação In\_\_\_\_\_. **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.

NASA - **National Aeronautics and Space Administration**, 2006. Disponível em: <[http://www.nasa.gov/exploration/thismonth/this\\_month\\_september.html](http://www.nasa.gov/exploration/thismonth/this_month_september.html)>. Acesso em: 18 dez. 2009.

NASA - **National Aeronautics and Space Administration**. 2008. Disponível em: <<http://www.grc.nasa.gov/WWW/K-12/airplane/airplane.html>>. Acesso em: 18 dez. 2009.

NAHMIAS, S. **Production and operations analysis**. 4 ed., Mcgraw-Hill: New York, 2008.

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

OLMOS, M. **Presidente da Airbus planeja parceria com a Embraer**. Disponível em: <[http://www.defesanet.com.br/eads/br\\_10.htm](http://www.defesanet.com.br/eads/br_10.htm)>. 2008. Acesso em 13 jul. 2009.

OTTOBONI, J. **China endurece para comprar da Embraer**. 2007. Disponível em <<http://www.defesanet.com.br/emb1/cn.htm>>. Acesso em 13 jul. 2009.

**PILATUS**. Disponível em: <[http://www.pilatus-aircraft.com/html/en/services/index\\_1137.asp](http://www.pilatus-aircraft.com/html/en/services/index_1137.asp)>, Acesso em: 29 dez. 2009.

PINEDO, M. L. **Planning and scheduling in manufacturing and services**. New York: Springer, 2005.

PINEDO, M. L. **Scheduling: theory, algorithms and systems**. 3 ed., New-Jersey: Prentice-Hall, 2008.

POTTS, C. N., STRUSEVICH, V. A. Fifty years of scheduling: a survey of milestones. **Journal of the operational research society**, vol. 60, pág 41-68, 2009.

QUADT, D. **Lot sizing and scheduling for flexible flow lines**. Springer: Berlin, 2004.

RAY, S., STAROBINSKI, D., CARRUTHERS, J. B. Performance of wireless networks with hidden nodes: a queuing-theoretic analysis. **Computer Communications**, vol. 28, pág. 1179-1192, 2005.

RILEY, F. J., **Assembly automation: a management handbook**. New York: Industrial Press, 1996.

RODAMMER, F. A., WHITE, K. P. Recent Survey of production scheduling. **IEEE Transactions on Systems, Man, and Cybernetics**, vol. 18, n. 6, nov./dez., 1988.

SANTOS, I. C. S., NETO, J. A., Gestão do conhecimento em indústria de alta tecnologia. **Produção**, v. 18, n. 3, set./dez. 2008, pág. 569-582, 2008.

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. M. R., MICHELON, P., ARENALES, M. N., SANTOS, R. H. S. Crop rotation scheduling with adjacency constraints. **Annals of Operations Research**, nov., 2008.

SCHÄFFER, M. W. Scheduling with forbidden sets. **Discrete Applied Mathematics**, vol. 72, pág. 155-166, 1997.

SCHEMENNER, R. W., Dealing with technological change. In:\_\_\_\_\_. **Production/operation management** – concepts and situations. 4 ed., New York: MacMillan Publishing Company, 1990. p. 674-706.

SCOTT, H. A. Modeling aircraft assembly operations. **Proceedings of the 1994 Winter Simulation Conference**, 1994

SECCHIN, V. M. S. **Implicações da organização da produção e do trabalho na atividade dos montadores de montagem estrutural**. 2007, 141 f. Dissertação (Mestrado em Engenharia de Produção) Departamento de Engenharia de Produção, Universidade Federal de São Carlos, São Carlos, 2007.

SEITZ, F., STEELE, L. W. **The competitive status of the U.S. civil manufacturing industry: a study of the influences of technology in determining international industrial competitive advantage.** U.S. Civil Aviation Manufacturing Industry Panel, National Academy of Engineering and the Commission on Engineering and Technical Systems, Washington D.C., National Academy Press, 1985.

SHEU, J-P., TU, S-C, HSU, C-H. Location-free topology control protocol in wireless ad hoc networks. **Computer Communications**, vol. 31, pág. 3410-3419, 2008.

SLACK, N., CHAMBER, S., JOHNSTON, R. **Administração da Produção.** 2 ed., Ailton Bonfim Brandão (trad.) São Paulo: Atlas, 2002.

SOUZA, T. O. **Construção de formalização do saber em montagem estrutural de aeronaves.** 2008. 180 f. Tese (Doutorado em Engenharia de Produção) Departamento de Engenharia de Produção, Universidade Federal de São Carlos, São Carlos, 2008.

STANFORD **Aircraft structural design.** Disponível em: <<http://adg.stanford.edu/aa241/structures/structuraldesign.html>>. Acesso em: 21 dez. 2009.

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.

U.S.A.. U.S. Department of Transportation. Federal Aviation Administration. **Pilot's handbook of aeronautical knowledge.** FAA-H-8083-25, 2003. Disponível em:< [http://www.americanflyers.net/aviationlibrary/pilots\\_handbook/chapter\\_1.htm](http://www.americanflyers.net/aviationlibrary/pilots_handbook/chapter_1.htm)>. Acesso em: 21 de dez 2009.

U.S.A.. U.S. Department of Transportation. Federal Aviation Administration. **Pilot's handbook of aeronautical knowledge.** FAA-H-8083-25A, Oklahoma City, 2008. Disponível em: <[http://www.faa.gov/library/manuals/aviation/pilot\\_handbook/](http://www.faa.gov/library/manuals/aviation/pilot_handbook/)>. Acesso em: 21 de dez 2009.

TEAL GROUP **World aviation market outlook.** 2006. Disponível em: <<http://www.leeham.net/filelib/PAC-AC-06.ppt>>. Acesso em: 9 de novembro de 2009.

TSERTOU, A., LAURENSSON, D. I. Revisiting the hidden terminal problem in a CSMA/CA wireless network. **IEEE Transactions on Mobile Computing**, vol. 7, n. 7, jul., 2008.

WÄSCHER, G.; HAUßNER, H.; SCHUMANN, H. An Improved Typology of Cutting and Packing Problems. **European Journal of Operational Research**, vol. 183, p1109-1130, 2007.

WEINTRAUB, A., ROMERO, C., BJØRNDAL, T., EPSTEIN, R. (Eds.) **International series in operations research & management science**: vol. 99. Handbook of operations research in natural resources. New York: Springer, 2007.

WOLSEY, L. A. **Integer programming**. New York: John Wiley & Sons, 1998.

YAMASHITA, D. S., ARMENTANO, V. A., LAGUNA, M. Scatter search for project scheduling with resource availability cost. **European Journal of Operational Research**, vol. 169, pág. 623-637, 2006.

YAMASHITA, D. S., ARMENTANO, V. A., LAGUNA, M. Robust optimization models for project scheduling with resource availability cost. **Journal of Scheduling**, vol. 10, pág. 67-76, 2007.

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**, vol. 27, n. 1, pág. 27-49, jan.-abr., 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 e Produção**, vol. 14, n. 13, pág. 545-555, set.-dez., 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**, vol. 5, n. 4, 2009.

YICK, J., BISWANATH, M., GHOSAL, D. Wireless sensor network survey. **Computer Networks**, vol. 52, pág. 2292-2330, 2008.

## 8 APÊNDICE A - Tabelas de durações das operações de montagem

Tabela 5: Duração (em u.t.) das operações das partes do subconjunto para o exemplo 1

Subconjunto	Peça 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	12	15	-
1	1	2	1	15	48	1
2	1	3	1	12	15	-
2	1	4	1	15	48	3
1	2	5	2	18	32	-
1	2	6	2	10	56	5
2	2	7	2	18	32	-
2	2	8	2	10	56	7
1	3	9	3	18	28	-
1	3	10	3	10	56	9
2	3	11	3	18	28	-
2	3	12	3	10	56	11
1	4	13	4	18	30	-
1	4	14	4	18	30	13
1	4	15	4	9	12	14
2	4	16	4	18	30	-
2	4	17	4	18	30	16
2	4	18	4	9	12	17
1	5	19	5	18	30	-
1	5	20	5	18	30	19
1	5	21	5	9	12	20
2	5	22	5	18	30	-
2	5	23	5	18	30	22
2	5	24	5	9	12	23
1	6	25	6	18	36	-
1	6	26	6	12	24	26
2	6	27	6	18	36	-
2	6	28	6	12	36	28
1	7	29	7	18	36	-
1	7	30	7	12	24	30
2	7	31	7	18	36	-
2	7	32	7	12	24	32
1	8	33	8	18	40	-
1	8	34	8	12	72	34

2	8	35	8	18	40	-
2	8	36	8	12	72	36

Tabela 6: Duração em unidades de tempo das partes do subconjunto 1 para o exemplo 2

Subconjunto	Peça 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	5	7	-
1	1	2	1	4	47	1
2	1	3	1	5	7	-
2	1	4	1	4	47	3
1	2	5	2	15	29	-
1	2	6	2	9	71	5
2	2	7	2	15	29	-
2	2	8	2	9	71	7
1	3	9	3	11	40	-
2	3	10	3	11	40	-
1	4	11	4	12	40	-
1	4	12	4	6	40	11
2	4	13	4	12	40	-
2	4	14	4	6	40	13
1	5	15	5	8	30	-
1	5	16	5	6	40	15
2	5	17	5	8	30	-
2	5	18	5	6	40	17
1	6	19	6	12	17	-
1	6	20	6	6	40	19
2	6	21	6	12	17	-
2	6	22	6	6	40	21
1	7	23	7	7	12	-
1	7	24	7	7	30	23
2	7	25	7	7	12	-
2	7	26	7	7	30	25
1	8	27	6	10	61	-
1	8	28	6	12	66	27
2	8	29	7	10	61	-
2	8	30	7	12	66	29