

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

MATEUS DE OLIVEIRA PILEGGI

OTIMIZAÇÃO DO SEQUENCIAMENTO DE
ATIVIDADES DE MANUTENÇÃO DE
AERONAVES COM ABORDAGENS DE
PROGRAMAÇÃO DE PROJETOS

SÃO CARLOS - SP

2019

MATEUS DE OLIVEIRA PILEGGI

OTIMIZAÇÃO DO SEQUENCIAMENTO DE ATIVIDADES DE MANUTENÇÃO DE
AERONAVES COM ABORDAGENS DE PROGRAMAÇÃO DE PROJETOS

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal de São Carlos, para obtenção do título de Mestre em Engenharia de Produção.

Orientador: Prof. Dr. Reinaldo Morabito

São Carlos - SP

2019



UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Engenharia de Produção

Folha de Aprovação

Defesa de Dissertação de Mestrado do candidato Mateus de Oliveira Pileggi, realizada em 11/07/2019.

Comissão Julgadora:

Prof. Dr. Reinaldo Morabito Neto (UFSCar)

Prof. Dr. Cleber Damiano Rocco (UNICAMP)

Profa. Dra. Vitória Maria Miranda Pureza (UFSCar)

Profa. Dra. Denise Sato Yamashita (UFSCar)

Aos meus pais João (in memoriam) e Lúcia Helena

Às minhas filhas, Valentina e Marília

AGRADECIMENTOS

Agradeço aos meus pais João Batista e Lúcia Helena, que me proporcionaram a oportunidade de precisar apenas estudar e me ensinaram a importância da educação formal. Ao exemplo de meu pai que concluiu o ensino superior na mesma idade em que eu concluo esta dissertação.

Agradeço à minha esposa Camila e minhas filhas Valentina e Marília pelo apoio nos momentos em que para concretização desse trabalho eu não pude estar presente como eu gostaria.

Agradeço aos meus irmãos Flávio, Simone e Rodrigo, pessoas realmente presentes em minha vida.

Agradeço ao meu orientador Prof. Reinaldo Morabito pelos ensinamentos, pelo sempre presente clima de parceria e pela compreensão nos momentos de dificuldade. Para mim uma referência como educador e pessoa.

Agradeço ao amigo Bruno, pela ajuda na compreensão do problema e delineamento deste trabalho lá em 2015, e pelos inúmeros momentos de dúvida em que pude contar com seu apoio.

Agradeço por fim, à Empresa que colaborou para a realização desse trabalho, suporte sem o qual ele não se concretizaria.

RESUMO

O presente trabalho tem como objeto de estudo o sequenciamento de atividades de manutenção aeronáutica, que pode ser visto como um problema de programação da produção. O trabalho é desenvolvido com a colaboração de um Centro de Manutenções Aeronáuticas, de médio porte, localizado no Brasil, que cuida de uma frota de dezenas de aeronaves. O estudo de caso tem por objetivo desenvolver, propor e aplicar dois modelos de otimização para representarem apropriadamente o problema, utilizando abordagens de programação de projetos. Estas abordagens são baseadas em Problemas de Programação de Projetos com Recursos Restritos (*Resource Constrained Project Scheduling Problems*), já explorados na literatura em outros contextos. Pretende-se contribuir para aprimorar o sequenciamento das atividades de manutenção aeronáutica e analisar a eficácia dos modelos propostos para resolver esse problema. Resultados computacionais obtidos da aplicação do modelo para exemplos realistas do estudo de caso são apresentados e analisados, mostrando o potencial dessa abordagem para representar e otimizar a manutenção aeronáutica. Os resultados mostram que ambos os modelos possuem possibilidade de aplicação prática, cada um com características particulares relacionadas à sua função objetivo, gerando resultados melhores se comparados à prática vigente.

Palavras chave: manutenção aeronáutica, indústria aeronáutica, aviação, sequenciamento de atividades, programação da produção, programação de projetos, *makespan*, *leadtime* médio.

ABSTRACT

The present work analyses the aircraft maintenance activities scheduling that can be seen as a production scheduling problem. The work is developed with the collaboration of a mid-size maintenance center located in Brazil that maintains a fleet of dozens of aircrafts. The case study aims to develop, propose and apply two optimization models that appropriately represents the problem using project scheduling methods. These methods are based in the Resource Constrained Project Scheduling Problems, already explored in other industrial settings. The aim is to contribute to enhance the aircraft maintenance activities scheduling and analyze the efficiency of the proposed models to solve the problem. Computational results obtained from the model application to real instances of the case study are presented and analyzed, showing the potential of this kind of approach to represent and optimize the aircraft maintenance. The results show that both models have the potential for practical application, each one with its objective function characteristics, achieving better results when compared to the case study practice.

Key-words: aircraft maintenance, aeronautical industry, aviation, activities sequencing, production scheduling, project scheduling, makespan, average leadtime.

LISTA DE FIGURAS

Figura 1.1: Custos e despesas do setor aéreo em 2017 (ANAC, 2018).....	11
Figura 2.1: Pacotes de manutenção ao longo do tempo.	29
Figura 2.2: Sequência lógica básica de atividades de manutenção.	31
Figura 2.3: Ilustração de uma aeronave de pequeno porte (voetla.org).	33
Figura 2.4: Cauda de um Boeing 747 (wikipedia.org).	33
Figura 2.5: Hangar de manutenção Lufthansa Technik em Frankfurt (airwaysmag.com).	34
Figura 3.1: Rede de precedência do processo de manutenção.	46
Figura 4.1: Rede de precedências para os problemas com uma aeronave.....	54
Figura 4.2: Gráfico de Gantt da solução encontrada para a instância com uma aeronave, teste 1.1 (minimizar o <i>leadtime</i> médio, $a_I = 5$).	54
Figura 4.3: Gráfico de Gantt da solução encontrada para a instância com uma aeronave, teste 1.2 (minimizar o <i>leadtime</i> médio, $a_I = 1$).	55
Figura 4.4: Rede de precedências para os problemas com duas aeronaves.....	56
Figura 4.5: Gráfico de Gantt da solução encontrada para a instância com duas aeronaves, teste 2.1 (minimizar o <i>leadtime</i> médio, $a_I = 5$).	57
Figura 4.6: Gráfico de Gantt da solução para a instância com duas aeronaves em série, teste 2.2 (minimizar o <i>leadtime</i> médio, $a_I = 1$).	58
Figura 4.7: Rede de precedências para os problemas com quatro aeronaves.....	59
Figura 4.8: Gráfico de Gantt da solução para a instância com quatro aeronaves em paralelo, teste 3.1 (minimizar o <i>leadtime</i> médio, $a_I = 5$).	61
Figura 4.9: Ocupação da mão de obra no teste 3.1.....	62
Figura 4.10: Ocupação da mão de obra no teste 3.2b.....	62
Figura 4.11: Solução esperada para o experimento teste 3.2.	63
Figura 4.12: Gráfico de Gantt da solução para a instância com quatro aeronaves em série, teste 3.2b (minimizar o <i>leadtime</i> médio, $a_I = 1$).	65
Figura 4.13: Gráfico comparativo, aumento do tamanho do problema versus aumento do <i>leadtime</i> médio.....	80
Figura 4.14: Gráfico comparativo, aumento do tamanho do problema versus aumento do <i>makespan</i>	80
Figura 4.15: Dispersão dos valores dos <i>gaps</i> de otimalidade.....	81
Figura 4.16: Gráfico de Gantt da solução do experimento 7.7.....	84
Figura 4.17: Gráfico de Gantt da solução do experimento 8.7.....	85
Figura 4.18: Gráfico de Gantt do experimento 8.34 sem a priorização da aeronave 3.	86
Figura 4.19: Gráfico de Gantt do experimento 8.34 com a priorização da aeronave 3.	86

LISTA DE TABELAS

Tabela 2.1: Natureza das tarefas de manutenção.	27
Tabela 4.1: Principais características dos experimentos de teste.	53
Tabela 4.2: Resultados dos experimentos de teste com função objetivo minimizar o <i>leadtime</i> médio.	66
Tabela 4.3: Resultados dos experimentos de teste com função objetivo minimizar o <i>makespan</i>	66
Tabela 4.4: Dados de entrada do problema com dados reais.	69
Tabela 4.5: Resultados dos experimentos função objetivo min <i>leadtime</i> médio, modelo (13a)-(18). ..	72
Tabela 4.6: Resultados dos experimentos função objetivo min <i>makespan</i> , modelo (13b)-(18).....	73
Tabela 4.7: Comparativo entre os valores das funções objetivo de cada modelo e seus equivalentes..	75
Tabela 4.8: Resultados dos experimentos 7.31 e 8.34, executados por 10h.....	77
Tabela 4.9: Tabela comparativa de experimentos baseados na MOD ideal.....	78
Tabela 4.10: Período da entrega de cada aeronaves dos experimentos 7.7 e 8.7	82

LISTA DE SIGLAS

ANAC	Agência Nacional de Aviação Civil
ATA	Air Transport Association
BOM	Bill Of Materials
CCPM	Critical Chain Project Management
CPM	Critical Path Method
CTM	Controle Técnico de Manutenção
EE	Exhaustive Enumeration
ERP	Enterprise Resources Planning
ES	Evolution Strategies
FAA	Federal Aviation Administration
FC	Flight Cycles
FH	Flight Hours
FIFO	First In First Out
MILP	Mixed Integer Linear Programming
MMRACP	Multi Mode Resource Availability Cost Problem
MMRCPSP	Multi Mode Resource Constrained Project Scheduling Problem
MO	Months
MOD	Mão de Obra Direta
MRB	Maintenance Review Board
MRP	Material Resources Planning
MSG-3	Maintenance Steering Group - 3
PAC	Productive Activities Control
PSP	Project Scheduling Problem
RBAC	Regulamento Brasileiro da Aviação Civil
RCPSP	Resource Constrained Project Scheduling Problem
SA	Simulated Annealing
TCPSP	Time Constrained Project Scheduling Problem

SUMÁRIO

1	INTRODUÇÃO	11
1.1	OBJETIVOS	14
1.2	METODO DE PESQUISA	15
1.3	ORGANIZAÇÃO DO TRABALHO	17
2	REVISÃO DA LITERATURA E PROBLEMA DE ESTUDO.....	19
2.1	OTIMIZAÇÃO NA MANUTENÇÃO AERONÁUTICA.....	19
2.2	O PROBLEMA DA MANUTENÇÃO	24
2.3	MODELOS DE REFERÊNCIA.....	35
3	FORMULAÇÃO MATEMÁTICA DO PROBLEMA	46
4	EXPERIMENTOS COMPUTACIONAIS.....	52
4.1	EXPERIMENTOS DE TESTE	52
4.2	EXPERIMENTOS COM UM PROBLEMA REAL.....	68
5	CONCLUSÕES.....	87
	REFERÊNCIAS BIBLIOGRÁFICAS	90

1 INTRODUÇÃO

A Indústria Aeronáutica de maneira geral é pouco explorada academicamente no que tange à Engenharia de Produção. Pesados são os investimentos em pesquisas nas áreas clássicas da engenharia, como a mecânica e a elétrica, por exemplo, principalmente por essas áreas estarem mais diretamente ligadas aos produtos da Indústria Aeronáutica em si, o que faz com que qualquer inovação traga diferencial competitivo importante aos produtos em um mercado acirrado.

Historicamente, a Indústria Aeronáutica de transporte de passageiros, ou Indústria da Aviação, é fortemente pressionada em custos, sendo o principal deles o custo do combustível. Este é responsável por aproximadamente 30% dos seus custos operacionais (ANAC, 2018), seguido dos custos de arrendamento, seguro e manutenção de aeronaves (20,3%). No cenário internacional, esse aspecto empurra para cima os custos operacionais de cada voo, elevando assim o número mínimo de assentos preenchidos que cada voo precisa ter, de forma a cobrir o custo operacional da empresa no voo. No cenário nacional, isso é ainda mais agravado já que o combustível é cotado em dólar. Uma vez que são necessários cada vez mais passageiros por voo para cobrir os custos operacionais, a competição por passageiros leva as empresas a praticarem patamares mais baixos de preços, de maneira a buscar a máxima ocupação dos voos, e assim garantir sua sobrevivência e algum lucro. A Figura 1.1 abaixo detalha a divisão dos custos das empresas aéreas em 2017 no Brasil.

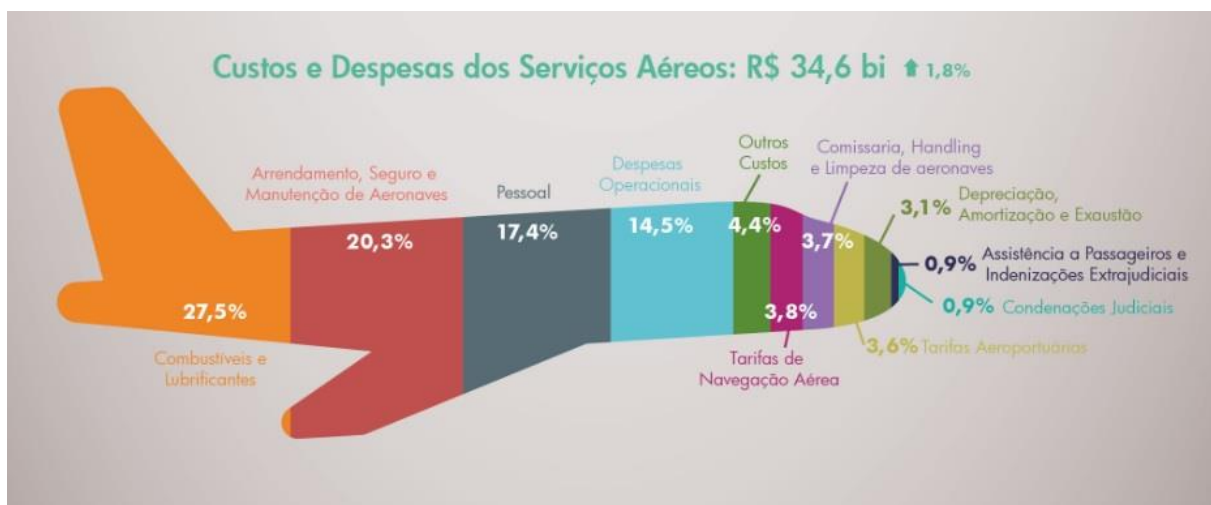


Figura 1.1: Custos e despesas do setor aéreo em 2017 (ANAC, 2018).

Similarmente à indústria de transporte de passageiros, o setor de transporte de cargas passa pelas mesmas dificuldades. Não é raro ver notícias de empresas aéreas com dificuldades financeiras, pedidos de concordata, ou ainda empresas que apresentam resultados ruins. Nesse cenário de pressão constante, principalmente, pelo preço dos combustíveis e competição acirrada, a indústria de aviação pressiona todo o restante da cadeia de custos em busca de eficiência, transferindo parte dessa pressão aos fabricantes, pois a eficiência das aeronaves torna-se fundamental na viabilidade econômica das empresas. É possível notar claramente uma tendência da Indústria Aeronáutica nesse sentido, buscando aeronaves cada vez mais eficientes.

Isso fica evidenciado ao se verificar a quantidade de materiais compósitos de fibra de carbono, mais leves que o alumínio, utilizados nas aeronaves, e que nas gerações mais recentes de aeronaves chegam a 50% da aeronave fabricada nesses materiais (Boeing, 2018). Existe ainda uma nova geração de motores chegando ao mercado, os *geared turbo fans*, motores que prometem entregar até 15% de eficiência no consumo de combustível (Pratt & Whitney, 2018), e em alguns casos 25% (Rolls Royce, 2018). Por fim, vale mencionar os esforços no *design* aeronáutico, em que as empresas buscam melhorias aerodinâmicas que possam proporcionar ganhos nessas características dos aviões. Alguns exemplos podem ser vistos em mudanças dos projetos das pontas de asas das aeronaves e no formato das próprias asas principalmente. Algumas dessas melhorias são apresentadas em Airbus (2017), Airway (2014), FlighGlobal (2013) e em Advanced Science News (2017).

Talvez toda essa competitividade também seja a origem do baixo interesse em pesquisas relacionadas à Engenharia de Produção, pois poderiam expor fragilidades, ou ainda, poderiam ser uma porta de entrada, uma brecha, que colocaria em risco os segredos competitivos, os diferenciais competitivos relacionados aos produtos. Mas é certo que as inovações mais diretamente ligadas ao produto em si chamam mais a atenção dos compradores, tendo um potencial muito grande de alavancar as vendas.

Dentro deste universo, na Indústria da Aviação, temos os chamados centros de serviço, que são responsáveis pela manutenção das aeronaves e têm o objetivo de manter em operação a frota de seus clientes. A indústria nesse ramo, similarmente aos demais, também é fortemente pressionada por resultados e divide-se de acordo com a característica do operador da aeronave. Existem, assim: a Aviação Comercial, a Aviação Executiva e a Aviação de Defesa.

Na chamada Aviação Comercial, têm-se o setor responsável pelo transporte de passageiros ao redor do mundo, muito fortemente pressionado em custos, que precisa maximizar o tempo de suas aeronaves fora do chão de maneira a garantir lucratividade. Via de

regra, não se permite possuir aeronaves “extras”, a exemplo do que ocorre em outras indústrias de transporte, de forma a suprir aeronaves em manutenção. Isso pressiona ainda mais os centros de serviços no sentido de minimizar as horas despendidas com a manutenção das aeronaves. Vale ressaltar aqui que, embora não seja regra, muitas vezes a empresa aérea na Aviação Comercial possui seus próprios centros de serviço de manutenção, dada a criticidade que essa operação tem para o seu negócio. Conforme mencionado antes, o peso dos custos da manutenção das aeronaves também pode ser visto na Figura 1.1, tendo ficado em 2017 na casa dos 20% dos custos operacionais.

Na Aviação Executiva, tem-se um cliente em geral exigente com prazos, e que apesar de voar significativamente menos se comparado à Aviação Comercial, requer que sua aeronave esteja disponível quando necessitar dela. Tal cliente pode ser um executivo que utiliza o recurso para gerir seu negócio, ou um cliente que demanda o produto para uso pessoal ou lazer. Este estudo utilizará dados reais de manutenção vindos de uma frota de aeronaves executivas.

Por fim, existe a Aviação de Defesa, que em geral faz uso severo de seu equipamento e demanda altos níveis de prontidão, de forma a garantir a proteção de seus territórios e patrimônios. Na Aviação de Defesa, os operadores são geralmente forças armadas. Esses operadores, relutam em delegar a manutenção de suas aeronaves a terceiros, seja para evitar acesso não autorizado a tecnologias bélicas, seja para não reduzir o número de aeronaves disponíveis. Outro aspecto importante aqui é a prevenção de sabotagem. Existem também alguns casos, esporádicos, de terceirização da manutenção desses tipos de aeronaves em casos específicos.

Por esses motivos, é possível observar alguma movimentação dessa indústria no sentido de buscar soluções para tais problemas, melhorando no sentido de explorar os conhecimentos e soluções que a Engenharia de Produção pode oferecer. Essa movimentação vai na direção de otimizar recursos, sejam eles humanos ou equipamentos, buscando propiciar aos clientes o menor tempo possível de aeronave parada, e, ao negócio, uma redução de custos operacionais.

A presente pesquisa conta com a importante colaboração de um Centro de Manutenção de Aeronaves localizado no Estado de São Paulo, com um time de mais de 250 funcionários. Esse centro cuida atualmente de uma frota de algumas dezenas de aeronaves, enfrenta um crescimento significativo e atende aeronaves que podem ser da Aviação Executiva ou da Aviação Comercial. Atende clientes que podem ser regulares ou esporádicos, mas na maior parte das vezes são regulares. Esse centro, embora possua um processo de

sequenciamento das atividades, em geral não o faz de maneira padronizada ou automatizada. Essa atividade é executada por pessoas com grande vivência e experiência no meio aeronáutico, e embora possuam grande conhecimento tácito e específico nesse setor, às vezes não são capazes de garantir ganhos mensuráveis.

1.1 OBJETIVOS

O presente trabalho investiga e explora uma lacuna na literatura científica em relação a trabalhos de otimização da manutenção aeronáutica. Também investiga a lacuna desta indústria em relação à aplicação de técnicas mais refinadas vindas da academia e as oportunidades de aplicação destas técnicas para uma empresa brasileira. Segundo Van Den Bergh (2013), a literatura sobre manutenção de aeronaves está claramente em menor número dentro da área de pesquisa operacional. Neste *survey* o autor revisa uma centena de trabalhos relacionados à manutenção de aeronaves e à pesquisa operacional, com o objetivo de facilitar o encontro de trabalhos em determinadas áreas de interesse na manutenção de aeronaves.

O problema da manutenção de aeronaves, de forma resumida, consiste na execução de uma série de atividades de inspeção, reparos e trocas de componentes. Essas atividades devem seguir uma certa ordem ou sequência. Segundo Arenales et al. (2015), esse problema, dentro da Engenharia de Produção, pode ser visto como um problema de programação da produção. Tal problema é de nível operacional e trata da decisão de designação de cada uma das tarefas a uma máquina, ou seja, a sequência de processamento dessas tarefas, bem como o instante de início e fim de cada uma delas. Ainda segundo os autores, o problema também se encaixa na classe de problemas de programação de projetos, pois trata de determinar uma sequência para um conjunto de atividades.

Neste estudo de mestrado pretende-se abordar e explorar alternativas de aprimoramento para a manutenção de aeronaves no cenário acima descrito, buscando reduzir a aplicação de recursos, ou ainda equilibrar esta aplicação de recursos. Pretende-se fazer isso estudando o sequenciamento das atividades de manutenção, aprimorando esse sequenciamento por meio da resolução de um modelo matemático de otimização baseado em modelos de programação de projetos, objetivando dessa forma, mostrar que é possível obter ganhos a partir de um sequenciamento estruturado das atividades de manutenção.

Em suma, o presente trabalho tem o objetivo de desenvolver, propor e aplicar dois modelos de otimização para representar apropriadamente o problema de manutenção de

aeronaves, utilizando abordagens de programação de projetos. Estas abordagens são baseadas no *Resource Constrained Project Scheduling Problem* (RCPSP), já explorado na literatura em outros problemas, e inicialmente introduzido por Pritsker et al. (1968). O objetivo aqui é contribuir para aprimorar o sequenciamento das atividades de manutenção e analisar a eficácia desse tipo de modelo para esse problema. Resultados computacionais obtidos da aplicação do modelo para exemplos realistas do estudo de caso são aqui apresentados e analisados.

A quantidade de mão de obra é um recurso restrito nos centros de manutenção e, constantemente, a sua escassez gera impactos no andamento das atividades. O estudo da alocação da mão de obra está no escopo deste trabalho. A linha de pesquisa é a de restringir os recursos de mão de obra direta (MOD) disponíveis e avaliar os impactos no *makespan* e no *leadtime* médio, buscando melhorá-los. O *makespan* é o instante de término de processamento de todas as atividades na manutenção das aeronaves (ARENALES et al., 2015), e o *leadtime* é o tempo de atravessamento total permitido para as aeronaves no processo de manutenção (SLACK et al., 2009).

1.2 METODO DE PESQUISA

O presente trabalho utiliza a abordagem quantitativa, que segundo Martins (2012), é a prática mais tradicional da ciência e com a qual todo aluno tem contato em algum momento. Pois, seja do ensino fundamental, médio ou superior, envolverá situações em que fará medições de algumas variáveis. Ainda segundo Martins (2012, p.47), “o ato de mensurar variáveis de pesquisa é a característica mais marcante da abordagem quantitativa”. De acordo com Bryman (1989), a abordagem quantitativa tem as seguintes preocupações: mensurabilidade; causalidade; generalização; replicação.

Talvez a principal característica da abordagem quantitativa, a mensurabilidade, visa testar as hipóteses por meio da medição e acompanhamento do comportamento das variáveis. As variáveis devem ser bem definidas de maneira a permitirem a validação das hipóteses. No presente trabalho, pretende-se otimizar o *makespan* e o *leadtime* médio da manutenção, e as variáveis que influenciam a função objetivo são estabelecidas e mensuráveis. Alguns exemplos são a quantidade de mão de obra aplicada, a disponibilidade de recursos e o tempo de preparação.

A causalidade segue diretamente ligada à mensurabilidade, e visa a garantir a relação de causa e efeito entre as variáveis a serem medidas e a variável dependente do

problema. Para o problema aqui abordado, basta observarmos que o *makespan* ou o *leadtime*, variáveis dependentes, têm relação direta com a mão de obra, variável independente, aplicada nas atividades. Uma vez que esta seja, por exemplo, elevada, espera-se que o avião tenha seu *leadtime* de manutenção e o *makespan*, reduzidos.

A generalização deve preocupar-se que o problema possa ser extrapolado para fora do objeto de pesquisa. Neste trabalho, sabe-se que a hipótese de solução do problema poderá ser levada para além do escopo de alguns pacotes pequenos de manutenção de um determinado modelo de aeronave. Será certamente possível extrapolar para diversos pacotes de manutenção desse mesmo modelo de aeronave e, ainda, para outros modelos de aeronaves.

A replicação, por sua vez, aborda a necessidade de que um projeto de pesquisa quantitativo deve ter a possibilidade de ser reproduzido, de maneira que permita a outro pesquisador poder obter resultados coerentes. No presente trabalho, isto é obtido, pois se trata da proposta e aplicação de um modelo matemático, que terá as equações descritas. O método de solução é descrito juntamente com os dados de entrada, o que permite que outro pesquisador altere os dados de entrada e obtenha seus próprios resultados.

Ressalta-se que na abordagem qualitativa, o pesquisador busca elaborar o seu modelo de solução levando em consideração as perspectivas dos indivíduos, e como eles enxergam e interpretam o problema. Outro aspecto importante é que se considera a realidade do ambiente em que este indivíduo está inserido e, conseqüentemente, o ambiente do problema trazido pela realidade organizacional. A pesquisa qualitativa preocupa-se com o fator subjetivo inserido na compreensão da realidade dos indivíduos que fazem parte da pesquisa. Para tal aceita um determinado nível de “desorganização” de maneira a contemplar essa subjetividade. Nada disso significa menor rigor de pesquisa, o que torna o controle da pesquisa ainda mais complexo. Na pesquisa quantitativa isso se contrapõe diretamente à mensurabilidade, não sendo relevante. O presente trabalho não irá levar em consideração os aspectos acima mencionados sobre pesquisa qualitativa, o que o afasta da abordagem qualitativa.

Vale entretanto ressaltar que existe certamente uma valorosa oportunidade de pesquisa qualitativa no objeto de pesquisa deste trabalho. O ambiente é desafiador com inúmeros fatores de pressão, situações altamente complexas e constantes mudanças de cenários, sendo que isso tudo afeta as pessoas envolvidas no processo e sua percepção. Essa é certamente uma oportunidade de pesquisa neste setor.

O método abordado neste trabalho é a Modelagem e Simulação, em que se busca reproduzir a realidade por meio de modelos matemáticos ou de simulação, de maneira a poder testar os resultados que podem ser obtidos, sem precisar enfrentar todas as dificuldades que se

teria para alterar efetivamente a realidade do objeto de pesquisa. Muitas vezes isso pode nem sequer ser desejável, pois poderia envolver riscos à operação. Talvez daí venha a principal vantagem deste método, pois ele permite a experimentação e teste de um sem número de possíveis soluções eliminando riscos. Vale destacar que este método foi fortemente impulsionado pelo desenvolvimento da computação, que, com computadores cada vez mais poderosos, permite a análise de problemas cada vez maiores e mais complexos. Mais detalhes do método de Modelagem e Simulação, podem ser encontrados em Nakano (2012) e Morabito e Pureza (2012).

O presente trabalho faz uso desse método, com a principal intenção de poder prever o que irá acontecer, uma vez utilizada a solução proposta. É utilizada a pesquisa de modelagem empírica, pois se busca um modelo que represente apropriadamente a realidade observada do objeto de trabalho. Segundo Morabito e Pureza (2012), a pesquisa denominada axiomática é primariamente dirigida a modelos de problemas idealizados. Além de pesquisa empírica ou axiomática os autores classificam ainda a Modelagem e Simulação como pesquisa normativa ou descritiva.

A pesquisa é qualificada como normativa, pois busca explorar soluções para o problema de programação das tarefas de manutenção, e, por fim, deixar claro qual a melhor decisão para a solução do problema, normatizando a sequência de atividades a ser executada para cada pacote de manutenção. Conforme Morabito e Pureza (2012, p.183): “ diferentemente da pesquisa descritiva, essa pesquisa baseia-se em modelos que prescrevem uma decisão para o problema, podendo ser baseada em modelos de otimização (ou programação) matemática. ” Em alternativa a essa opção, seria possível utilizar a modelagem empírica descritiva, buscando apenas reproduzir e descrever a realidade da operação do objeto de estudo, permitindo melhor estudá-lo. Entende-se que essa opção seria mais adequada a um modelo de simulação, dada a realidade do problema em estudo, o que não será explorado no presente trabalho.

1.3 ORGANIZAÇÃO DO TRABALHO

O trabalho está organizado da seguinte forma. Este primeiro capítulo apresenta brevemente a Indústria da Aviação e introduz o problema deste estudo, explica as motivações do trabalho e seus objetivos. Ainda neste capítulo é apresentada uma justificativa da escolha do método de pesquisa.

O segundo capítulo analisa alguns trabalhos de otimização de manutenção na Indústria da Aviação. O capítulo explica o problema deste estudo em detalhes, mostra algumas simplificações adotadas de forma a permitir a análise do problema, e, é concluído com a apresentação dos principais modelos de referência que serviram de base para a formulação dos modelos matemáticos desse trabalho.

O terceiro capítulo apresenta a formulação matemática dos dois modelos propostos nesse trabalho, baseados nas formulações de outros estudos de gerenciamento de projetos, e adaptadas para o presente estudo. O quarto capítulo apresenta os resultados dos experimentos computacionais: na primeira parte, são mostrados os resultados de uma série de experimentos que serviram de teste para os modelos, e, na segunda parte, os resultados de diversos experimentos construídos sob dados reais do estudo de caso.

O quinto e último capítulo apresenta as conclusões do trabalho e delinea as perspectivas futuras de pesquisa.

2 REVISÃO DA LITERATURA E PROBLEMA DE ESTUDO

Neste capítulo, inicialmente, se revisa a literatura relacionada com otimização na manutenção aeronáutica. Em seguida descreve-se o problema de manutenção aeronáutica com base na literatura e também no estudo de caso realizado na empresa brasileira. O capítulo é concluído com uma revisão dos modelos de referência que serviram de ponto de partida para a formulação do modelo do Capítulo 3.

2.1 OTIMIZAÇÃO NA MANUTENÇÃO AERONÁUTICA

Um dos grandes problemas no setor aeronáutico é a manutenção dentro da malha de voos. Quando uma companhia aérea voa, ela busca maximizar a utilização da aeronave, isso significa mantê-la voando na maior parte do tempo. Isso faz com que as paradas para manutenção sejam um problema importante a ser resolvido. Talvez esse seja o problema mais tradicional na aviação: como alocar as paradas de manutenção de uma aeronave em seu calendário de voos? A aeronave não pode voar com nenhum componente ou inspeção vencidos, mas é desejável que ela pare o mais próximo possível desses vencimentos de forma a maximizar sua utilização. Não é também desejável programar um voo para essa aeronave que a leve para muito longe da base onde ela fará sua manutenção. Não faz sentido programar um voo, por exemplo aos Estados Unidos, para uma aeronave que na sequência deverá parar para realizar uma manutenção, se essa manutenção será feita em um centro de manutenções localizado, por exemplo, no Brasil.

O propósito do presente trabalho é diferente nessa linha. Ele buscará otimizar o processo de manutenção em si, passando a olhar a manutenção a partir do momento em que a aeronave chega ao centro de manutenção. Acredita-se que esse problema foi menos explorado que o problema da alocação das manutenções no calendário dos voos.

Saramanayake e Kiridena (2012) propõe uma estrutura integrada para o planejamento à frente das atividades de manutenção de aeronaves, levando em consideração incertezas. Os autores comentam que as empresas de manutenção de aeronaves em sua maioria, tem utilizado pacotes comerciais ERP/MRP (*Enterprise resource planning / Material requirements planning*) tradicionalmente utilizados na indústria de transformação. O uso desses sistemas não seria capaz de gerir a manutenção de aeronaves por tratar em blocos, e de forma descoordenada, os pacotes de planejamento das necessidades de materiais (MRP), método do

caminho crítico (CPM – *Critical Path Method*) e controle de atividades produtivas (PAC – *Productive Activities Control*).

Para os autores, essa tradicional estrutura não interligada de pacotes não funciona bem para a manutenção de aeronaves. A justificativa é que aproximadamente 50% das atividades em uma manutenção não são previstas, mas oriundas de inspeções realizadas. Tais inspeções geram necessidade de novas atividades, e essas novas atividades geram novas necessidades de materiais. Com um nível elevado de incertezas, o planejamento fica defasado e dessincronizado rapidamente. Por esse motivo, o planejamento de manutenção requer sistemas capazes de replanejamento a qualquer momento, não levando em consideração apenas o replanejamento da rede de atividades, mas também de todas as necessidades de materiais e os prazos desses materiais.

Os autores propõem a utilização desses mesmos meios tradicionais, no entanto, utilizando as BOM (*Bill of materials*) hierarquicamente ligadas, redes de atividades em malha fechada e roteiros das sequências de atividades. Isso é obtido por meio de uma estrutura de dados unificada, ou seja, com todas essas informações na mesma base de dados. Com essa integração, após qualquer perturbação na execução das atividades, uma realimentação pode ser feita ao sistema e um replanejamento completo pode ser feito. Tal replanejamento é o planejamento à frente já normalmente realizado, pelos sistemas MRP. A vantagem é a possibilidade do replanejamento sistêmico durante a execução do projeto, proporcionando uma maior consciência do andamento do projeto como um todo. Isso é capaz de permitir correções e soluções, e ainda pode fornecer prazos mais realistas.

Em um trabalho recente, Junqueira et al. (2018) fazem um estudo de caso em um centro de manutenções de aeronaves também no Brasil. O estudo relata a utilização e aplicação da teoria da corrente crítica CCPM (*Critical Chain Project Management*) juntamente com o método do caminho crítico CPM no gerenciamento de projetos, com o objetivo de reduzir o tempo necessário para execução da manutenção preventiva. Os resultados são significativos, indicando que uma manutenção, que os autores definem como leve (pequena), pode ser reduzida de 11 para 5 dias trabalhados. Mattioda (2002) e Srinivasan (2007), ambos aplicam os conceitos de CCPM na manutenção de grandes aeronaves de transporte da força aérea norte americana. Este último relata ainda a aplicação de conceitos de manufatura enxuta (*Lean Manufacturing*), conseguindo reduzir o tempo de uma revisão completa de 360 para 240 dias. Relata ainda que a quantidade de aeronaves revisadas ao ano aumentou de 17 para 20, e a quantidade de aeronaves em processo foi reduzida de 16 para 13. Esse resultado é significativo pois representa o aumento de 3 aeronaves disponíveis para operação.

Atli e Kahraman (2012) propõem minimizar o tempo de conclusão das atividades de manutenção utilizando um modelo de caminho crítico difuso. Nesse trabalho os autores utilizam a teoria do caminho crítico (CPM), e substituem os tempos de execução das tarefas, normalmente determinísticos ou probabilísticos, por um conjunto de tempos difusos trapezoidais, como uma forma de lidar com as incertezas.

A alocação adequada de mão de obra é um problema bastante explorado na literatura relacionada à manutenção de aeronaves. Nestes casos, geralmente o objetivo é gerar as listas de pessoal a serem alocados nos turnos de trabalho de forma a minimizar a quantidade de mão de obra necessária, garantindo disponibilidade de todos os tipos de mão de obra diferentes. Essa é uma realidade muito presente nas atividades de manutenção de linha, executadas nas aeronaves entre um voo e outro.

Um exemplo clássico desse tipo de estudo é o trabalho de Dijkstra et al. (1994). Neste trabalho os autores desenvolvem um sistema de suporte à decisão para a *KLM Royal Dutch Airlines*. Essa empresa, de origem holandesa, é uma das maiores da Europa e a mais antiga empresa aérea ainda em operação nos dias de hoje, possuindo uma grande divisão de manutenção que cuida da frota da própria companhia e presta serviços de manutenção a diversas outras empresas aéreas. Nesse trabalho o problema de otimização da mão de obra é trabalhado com um modelo de programação linear colocado dentro do sistema desenvolvido. São analisadas a carga de trabalho bem como a alocação das diferentes especialidades. Esse modelo é resolvido por meio de uma heurística que obtém soluções não ótimas rapidamente, mas consideradas boas para a empresa.

Alfares (1999) estuda alocação da mão de obra de manutenção das aeronaves pertencentes a uma empresa produtora de petróleo na Arábia Saudita. O cenário é de aumento de 3 aeronaves na frota e a empresa já está com excesso de horas extras no time de manutenção. O autor propõe um modelo de programação linear para sequenciar a mão de obra, que está dividida em dois turnos de trabalho por dia, mudando o período de programação de cinco dias para sete dias. Com os resultados do trabalho é possível reduzir a necessidade de horas extras e realizar todas as manutenções sem necessidade de aumentar o time de manutenção. Outros trabalhos sobre otimização da alocação e sequenciamento de mão de obra incluem Beliën et al. (2011), Bertsimas et al. (2014) e Butchers et al. (2001).

Ahire et al. (2000) propõe um algoritmo de ES (*Evolution Strategy*), para o problema de manutenção preventiva de aeronaves. Os autores definem o problema abordado como sendo um *work-force constrained preventive maintenance problem*, ou problema de manutenção preventiva com mão de obra restrita. Eles buscam minimizar o *makespan* das

atividades, sendo que todas as tarefas devem ser completadas para que a manutenção seja considerada concluída. O tempo não é restrito, mas sim o recurso mão de obra, que pode ser considerado como mono ou multifuncional. O algoritmo também pode ser aplicado a outras atividades complexas, como a manutenção de trens e navios.

Os autores comparam a performance de seu algoritmo diante de outros dois possíveis métodos de solução, enumeração completa (EE – *Exhaustive Enumeration*) e SA (*Simulated Annealing*). Em ambos os casos, os autores (AHIRE et al. 2000) mostram, ao realizarem 852 experimentos de teste, que o algoritmo proposto é mais eficiente. No primeiro caso, comparando com EE o algoritmo fica mais eficiente à medida que o tamanho dos problemas aumentam, pois EE precisa analisar todo o espectro de soluções factíveis. No segundo caso, a SA precisa analisar aproximadamente doze vezes mais soluções que ES. Os autores comentam ainda que o tempo computacional aumenta à medida que o problema cresce, e diminui com a utilização de mão de obra de multifuncional.

Quan et al. (2007), estuda o *trade-off* entre mão de obra e *makespan* na manutenção preventiva em equipamentos complexos. Em um centro de manutenções de equipamentos de alta complexidade como aviões, a mão de obra, via de regra, é mais cara, e portanto, é desejável minimizá-la. Ao mesmo tempo deseja-se retornar o equipamento ao serviço no menor tempo possível, o que acontece apenas após realizadas todas as tarefas de manutenção, o que é mais facilmente obtido com uma quantidade maior de mão de obra. Este *trade-off* busca o equilíbrio entre mão de obra e redução do *makespan*. Os autores tratam como um problema de sequenciamento multi-objetivo, em que se busca minimizar a mão de obra e o *makespan* ao mesmo tempo.

As atividades podem demandar dois tipos diferentes de mão de obra, e cada tipo de mão de obra possui apenas uma única especialidade. Para o início de cada atividade, toda a mão de obra necessária de cada uma das especialidades deve estar disponível. No entanto uma vez terminada cada atividade, de cada especialidade, a mão de obra é imediatamente liberada. Os autores utilizam um método de solução baseado em algoritmos evolutivos com a medida de qualidade das soluções (aptidão) sendo a abordagem ótima de Pareto. É proposta uma abordagem de seleção das soluções com melhor aptidão com base em preferências, o que pode levar a soluções mais alinhadas com as decisões no gerenciamento das atividades.

Safaei et al. (2011) propõe um modelo de programação inteira mista para sequenciar atividades de manutenção numa frota de aeronaves militares. O problema consiste em sequenciar atividades de manutenção nas aeronaves com o objetivo de maximizar o número de aeronaves em prontidão para cumprir uma programação diária e pré-determinada de voos.

Uma vez encontrada uma falha crítica, a aeronave é removida da linha de voos e colocada na fila para reparo. A principal restrição também é a mão de obra especializada.

Os autores resolvem o modelo matemático por meio do método *branch-and-bound* clássico. Para os autores os resultados com esse método são satisfatórios, sendo que os resultados em instâncias de testes baseados em dados reais indicaram que a quantidade de tarefas tem pouco impacto no esforço e tempo computacional. O que mais impactou tais parâmetros foram a quantidade de voos programados, as quantidades de mão de obra disponíveis e as diferenças entre as quantidades de cada tipo.

Cotreau (2015) desenvolve um trabalho de sequenciamento de atividades em grandes *checks* de manutenção no Canadá, aplicado em uma empresa de manutenção (*Jazz Aviation*). Trata-se de um modelo para sequenciar apenas uma aeronave de um determinado modelo com o objetivo de minimizar o *makespan*. Contudo, a autora menciona que o ideal seria conseguir sequenciar até quatro aeronaves ao mesmo tempo. O modelo é também baseado no clássico RCPSP, no entanto é utilizada uma heurística de três estágios, combinada com um modelo matemático, para resolvê-lo. O *check* de manutenção é dividido em três etapas e cada uma delas é otimizada, pelo modelo, separadamente. O principal recurso restrito abordado é a mão de obra.

A formulação utilizada é de um modelo baseado em eventos, que a autora afirma desempenhar melhor para problemas maiores. O problema é tratado como monomodo e a quantidade de recursos necessários à execução das tarefas é definida pelo controle de produção. As tarefas também foram agrupadas de forma a reduzir a quantidade de tarefas a serem sequenciadas. Inicialmente a autora pretendia trabalhar com 250 tarefas que, após agrupadas, foram reduzidas e aproximadas para apenas 100.

O presente trabalho também utiliza de um agrupamento de tarefas, que será detalhado na seção 2.2. A abordagem também é monomodo. Acredita-se que com a abordagem adotada é possível tratar problemas de variados tamanhos, e também cobrir um período de tempo satisfatório dentro do horizonte de planejamento, em um tempo computacional aceitável para a realidade de um centro de manutenção. Acredita-se ainda que é factível o uso de métodos exatos clássicos, dada a capacidade computacional hoje existente, dispensando prês e pós processamentos mais elaborados. Toda essa abordagem do problema e simplificações são detalhadas na seção seguinte (Seção 2.2).

Cotreau (2015) afirma, por fim, que os resultados iniciais de implementação apontam para um sequenciamento que vai de 95% de otimalidade em seis minutos, a até 99% de otimalidade em 20 min. A empresa estudada considera satisfatório resultados nesses níveis,

dado que a diferença no *makespan* não é significativa quando a otimalidade varia de 95 para 99%.

2.2 O PROBLEMA DA MANUTENÇÃO

Conforme mencionado no capítulo anterior, a Indústria da Aviação é um setor fortemente pressionado pela redução de prazos e custos. Estes pela necessidade de reduzir custos para absorver as constantes altas nos preços dos combustíveis e poder competir pelos passageiros, aqueles para maximizar o tempo das aeronaves voando. Não se pode perder de vista que uma aeronave parada significa que a empresa está deixando de vender passagens em sua malha de voo.

Aeronaves são equipamentos complexos, com uma ampla gama de sistemas automatizados e redundantes que não podem falhar, sob pena de diversas perdas, inclusive vidas, o que pode levar a mais sólida das empresas à falência. Por essa razão, todo avião possui um plano de manutenção preventiva, que deve ser seguido à risca pelo operador da aeronave, sendo fortemente fiscalizado pela autoridade aeronáutica do país onde a aeronave opera.

De forma geral, as paradas de manutenção em uma aeronave são determinadas em função do tempo de horas de voo, *flight hours* (FH), que aquela aeronave acumula com o passar do tempo, e da quantidade de ciclos de voo, *flight cycles* (FC), também acumulada com o passar do tempo. Ciclos de voo são a quantidade de decolagens e pousos de uma aeronave, sendo que um ciclo representa uma decolagem e um pouso. Além das horas de voo e ciclos, também existe um controle temporal de inspeções, que delimita que determinadas manutenções, ou *checks*, devem ser feitos na aeronave ainda que o limite de horas de voo ou ciclos não tenha sido atingido. Por exemplo, um plano de manutenção pode determinar que uma aeronave deva parar para cumprir determinadas atividades de manutenção, a cada 500 horas de voo ou 250 ciclos, ou ainda 6 meses, o que ocorrer primeiro.

Na manutenção de aeronaves, além do controle da aeronave em si, componentes importantes podem ter controle dedicado de manutenção, sendo os principais exemplos os motores e os trens de pouso. O controle desses componentes também é feito, a exemplo da aeronave, por meio de FH e/ou FC, que denomina tais equipamentos como componentes controlados. Em uma aeronave existem muitos deles. Outros exemplos seriam alguns computadores, geradores de energia, bombas hidráulicas, bombas de combustível, atuadores, algumas válvulas, entre outros.

Diz-se que componente controlado em um avião tem “vida própria”, pois esse componente pode transitar entre diferentes aeronaves, e deve ter um controle dedicado de suas manutenções. Nesse caso, pode-se ter componentes em uma aeronave que possuam um número maior de FH e FC que o próprio avião em si. Para simplificar a compreensão, basta pensar em uma aeronave que apresentou defeito em um dos seus motores, e que esse motor precisará ser enviado ao fabricante para ser reparado. Esse processo de reparo envolve enviar o motor ao fabricante, que pode estar localizado em outro país, mais o reparo e devolução do motor, processo que mesmo sendo extremamente eficiente levaria vários dias, provavelmente semanas. A aeronave ficaria parada esperando esse reparo? Não, devido ao alto custo de oportunidade, a empresa aérea simplesmente instala outro motor na aeronave, processo que leva apenas algumas horas, e coloca a aeronave de volta à malha de voo, voltando a gerar resultado para a empresa.

Embora os componentes de um avião em geral sejam bastante caros, frente ao prejuízo que gerariam ao deixarem uma aeronave parada à espera destes, as empresas aéreas possuem componentes adicionais para substituição rápida em caso de falha. Isso porque o seu prejuízo seria muito maior caso a aeronave ficasse parada à espera destes para poder voltar ao serviço.

Pode-se traçar um paralelo com a manutenção dos carros de forma a simplificar a compreensão. Assim como um veículo deve realizar sua revisão a cada 1 ano ou 10 mil quilômetros, uma aeronave terá que realizar sua revisão a cada 1 ano, 500 FH ou 250 FC. Vale ressaltar que os números mencionados são exemplos, não representando toda a gama de aeronaves e inspeções existentes. Normalmente uma aeronave possui atividades de manutenção que devem ser feitas diariamente, semanalmente, quinzenalmente, mensalmente, trimestralmente, semestralmente, anualmente, bianualmente e assim por diante.

O plano de manutenção de uma aeronave é bastante complexo, podendo ter, por exemplo, mil e quinhentas atividades diferentes e mais de uma centena de componentes controlados. Devido a esta complexidade, geralmente as empresas possuem uma área dedicada ao plano de manutenção, denominada Controle Técnico de Manutenção (CTM), responsável por acompanhar o controle de horas e ciclo de todas as atividades e componentes, e por informar os vencimentos que estão próximos.

Vale ressaltar que um controle tão pormenorizado vem de estudos estatísticos sobre os momentos de falhas dos componentes e estrutura da aeronave. Em um passado não tão distante, no começo da aviação, os componentes eram simplesmente trocados de forma a garantir a segurança da aeronave. Esse tipo de controle existe até hoje, são os componentes denominados *hard time*, ou componentes de vida útil controlada, que uma vez atingido um

determinado número de horas de voo ou ciclos, devem ser trocados ou revisados (reparado em condição de novo), não importando sua condição atual de operação. Ou seja, esse item ainda que operacional e em boas condições, deve ser substituído ou revisado. No início da aviação, toda a manutenção nas aeronaves era feita com base em vida útil controlada. No entanto, o controle da manutenção evoluiu no sentido do controle estatístico das falhas, e isso aconteceu justamente para reduzir os custos de manutenção, que são bastante significativos, já que a maioria dos componentes são caros. Um alto volume de trocas representa diretamente um custo igualmente elevado (FAA, 1978).

Essa evolução nos planos de manutenção das aeronaves com base em análises estatísticas das falhas nos componentes foi um importante passo no universo da manutenção de aeronaves, representando um marco muito significativo na redução dos custos. Essa filosofia surgiu em 1980, sendo utilizada ainda nos dias de hoje, e foi denominada MSG-3 (do inglês *Maintenance Steering Group-3*). Nada mais é do que uma técnica de avaliação em camadas das probabilidades de falhas e os riscos envolvidos com essas falhas dos sistemas das aeronaves ou de suas estruturas, culminando em uma diretriz principal de manutenção daquele projeto de avião. A metodologia tem foco em uma operação segura e barata. O resultado da aplicação da metodologia MSG-3 é um documento que passa a ser a linha mestre de manutenção daquele projeto de avião, sendo denominado MRB (do inglês, *Maintenance Review Board*), ou programa de manutenção (ATA MSG-3, 2002.1).

Quando uma empresa aérea compra uma aeronave, ela pode seguir o plano de manutenção sugerido pelo fabricante, ou ela pode customizar esse plano de acordo com as suas necessidades, de forma a favorecer sua operação. Para tal, esse plano deve ter como base o MRB da aeronave e deve ser aprovado junto à autoridade aeronáutica que certifica a operação da empresa. Uma vez aceito o plano proposto, essa mesma entidade fiscalizará a aplicação do plano aprovado. No Brasil, essa entidade é denominada ANAC, Agência Nacional de Aviação Civil.

Um *check* de manutenção de um avião, ou uma parada de manutenção, consiste em parar a aeronave para cumprir uma série de tarefas obrigatórias, conforme o plano de manutenção a ser seguido. Essas tarefas podem ser de seis naturezas diferentes: estrutura, mecânica, aviônicos, interiores, selagem e pintura. A Tabela 2.1 a seguir ilustra a natureza das tarefas de manutenção.

Tabela 2.1: Natureza das tarefas de manutenção.

	Atividades	Exemplos
Estrutura	Inspeção e reparos em estruturas aeronáuticas.	Avaliação de riscos e trincas em estruturas de alumínio, confecção de reparos etc.
Mecânica	Inspeção e troca de componentes mecânicos.	Remoção e instalação de rodas, troca de componentes da estrutura do trem de pouso, substituição de válvulas etc.
Aviônica	Testes dos sistemas eletrônicos, diagnósticos e troca dos componentes do sistema elétrico.	Troca de computadores de sistemas, testes de continuidade em cablagens etc.
Interiores	Desmontagem e reconfiguração do interior da aeronave.	Instalação de <i>sidewalls</i> , reparos em poltronas, cintos etc.
Selagem	Vedação das junções nas estruturas aeronáuticas.	Vedação de tampas dos tanques de combustível.
Pintura	Inspeção, remoção de pinturas e aplicação de tintas.	Remoção e repintura de componentes estruturais com pintura comprometida.

Assim como as tarefas de manutenção se dividem de acordo com sua natureza, no objeto de estudo deste trabalho, a mão de obra também se divide da mesma forma. Existem seis tipos de mão de obra na empresa estudada, a saber: técnicos em estruturas; técnicos em mecânica; técnicos em aviônicos; técnicos em interiores; técnicos em selagem; técnicos em pintura. Desta forma, cresce a complexidade do problema, pois não basta gerenciar um único tipo de MOD, mas vários.

As manutenções programadas podem ser ditas pequenas ou grandes, sendo que nas aeronaves objeto deste estudo, podem quando grandes chegar a mais de mil tarefas de manutenção, e quando pequenas possuem aproximadamente sessenta tarefas de manutenção. Em geral, no universo das companhias aéreas se utiliza uma classificação, que pode ser A, B, C e D, para indicar o tamanho das manutenções planejadas, sendo denominado *check A* para as menores manutenções e *check D* para as maiores manutenções.

Algumas manutenções pequenas podem ser feitas no portão de embarque da aeronave entre um voo e outro. Outras manutenções são mais longas e demandam a aeronave parada por algumas horas, podendo ser feita no chamado pernoite, em que a manutenção é feita durante a noite e a aeronave é novamente disponibilizada para voltar a voar na manhã seguinte. O pernoite é geralmente feito no próprio aeroporto com a aeronave sendo deslocada ao hangar de manutenção.

Checks A até *D* são maiores e precisam da aeronave parada por vários dias, demandam que a aeronave seja deslocada a uma das bases de manutenção da companhia, ou do prestador de serviço de manutenção. O *check D* é considerado o maior tipo de manutenção

contido no plano de manutenção. A aeronave é totalmente desmontada, seu interior é completamente removido, e suas estruturas e componentes totalmente inspecionados. Um *check D* se estende por semanas e dependendo do esforço de mão de obra alocado, pode facilmente durar próximo de um mês ou até pouco mais de um mês.

Via de regra, uma aeronave chega para manutenção com um universo de atividades conhecidas para serem executadas, denominado de manutenção planejada, e um universo desconhecido, trazido e reportado pela tripulação na reunião de entrega da aeronave, denominado manutenção não planejada. Neste universo das manutenções não planejadas, estas podem ser classificadas como “GO” e “NO GO” (“passa” e “não passa”), representando respectivamente, problemas com os quais a aeronave pode conviver e problemas que devem necessariamente ser solucionados durante a parada de manutenção que será iniciada.

As aeronaves da frota em análise neste estudo seguem um plano de manutenção denominado estruturado, recomendado pelo fabricante. Não se entrará neste estudo em detalhes que possam permitir a identificação tanto das aeronaves da frota estudada quanto da empresa de manutenção que aceitou colaborar com este trabalho, a fim de resguardar àqueles que colaboraram, e, proteger o sigilo de seus negócios. Esse plano de manutenção estabelece um pacote de atividades que deve ser realizado a cada período de tempo vencido em relação ao tempo de vida da aeronave.

O fabricante da aeronave estabelece esses pacotes de manutenção com base em um perfil de voo anual, ou seja, esses planos podem ser seguidos desde que o operador se adeque ao limite de horas de voo e ciclos estabelecidos. No caso estudado, o fabricante estabelece o plano 1 para clientes que irão voar no máximo 450 FH e 200 FC ao ano e no plano 2, o máximo de 900 FH e 450 FC ao ano. Ao utilizar esses planos, o cliente terá um pacote de atividades de manutenção que devem ser repetidas mensalmente, ou a cada um mês acumulado na idade da aeronave, um pacote de atividades de manutenção a cada três meses ou pacote 3MO (MO = *months*, do inglês meses), um pacote para 6MO, 12MO, 18MO e assim por diante até 192MO.

A partir da 6MO, o fabricante estabelece novos pacotes de manutenção em intervalos regulares de 6 meses. Outro ponto importante a notar é que quando uma aeronave chega a, por exemplo, 12 meses de idade, o pacote de manutenção a se realizar é o pacote de 12MO juntamente com seus múltiplos. Ou seja, com 12 meses uma aeronave deverá executar os pacotes: 12MO + 6MO + 3MO + 1MO. A ilustração abaixo mostra uma linha do tempo da idade de uma aeronave fictícia e seus pacotes de manutenção.

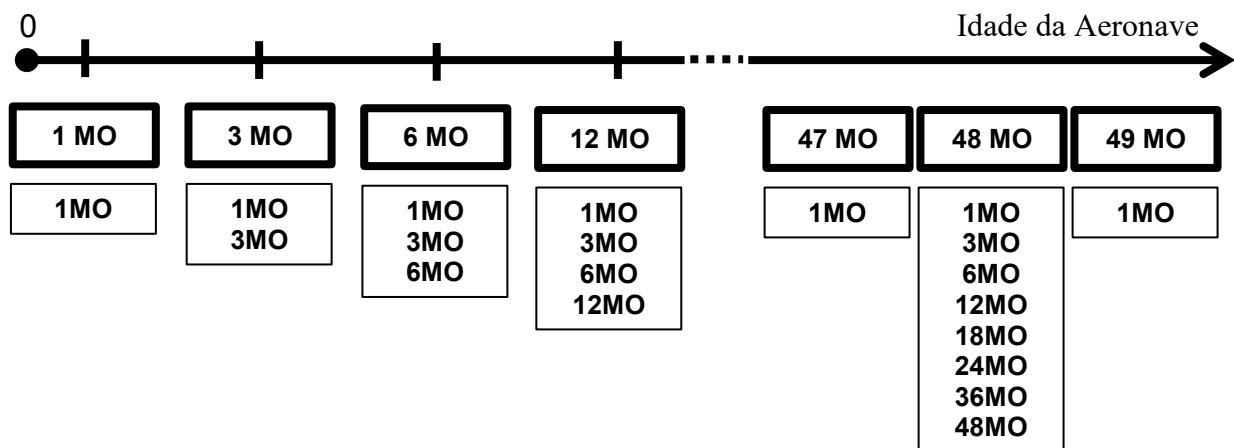


Figura 2.1: Pacotes de manutenção ao longo do tempo.

Vale notar na Figura 2.1 que a cada seis meses a aeronave passará pelas manutenções que são múltiplos de seus meses de idade, e que aos 48 meses de idade a aeronave deverá cumprir oito pacotes de manutenção. O resultado é um pacote grande, representando o que seria um *check C*. No entanto, aos 47 e 49 meses de idade os pacotes são pequenos, pois apenas o pacote de 1MO é múltiplo de 47 e 49.

De posse das informações mencionadas até agora, é possível entender como é montado um pacote ou *check* de manutenção da frota em estudo. Essas aeronaves param para realizar manutenções programadas de maior monta a cada seis meses, as manutenções de um mês e as de três meses podem ser consideradas *checks A* e são feitas pelo cliente em sua própria base de operações. Existem ainda para essa frota inspeções pré-voo, inter-voos, pós-voo, inspeções a cada 48 horas e a cada 14 dias ou 100 horas de voo. As inspeções pré, pós e inter-voos são muito rápidas e ocorrem sempre antes de um voo, depois de um voo e entre dois voos, respectivamente, conforme o próprio nome dessas manutenções deixa claro. As inspeções pré, pós e inter-voos são uma espécie de unanimidade entre os fabricantes, e existem para quase todas as aeronaves.

Conforme já mencionado, a quantidade de tarefas de uma manutenção programada pode variar de aproximadamente 50 tarefas até 1200 tarefas. É até razoável pensar em uma rede de atividades e precedência para um pacote de 50 ou 100 tarefas. No entanto, quando se atinge um patamar muito grande, como 1000 tarefas, isso pode ficar impraticável. Isso acontece pois, dificilmente um *check* é sempre igual para um mesmo período de vencimento, o que demandaria a construção da rede de precedências cada vez que um pacote de manutenção fosse definido.

Muito embora as atividades estejam agrupadas em pacotes, elas devem ser controladas individualmente, e podem ficar descasadas do seu período de manutenção esperado. Denomina-se essa tarefa descasada do plano de manutenção como tarefa “fora de fase” (*out of phase*). As tarefas *out of phase* são bastante frequentes em qualquer aeronave. O principal exemplo desse tipo de tarefa são os componentes de vida controlada, que por terem vida própria e poderem transitar entre diferentes aeronaves, raramente sofrem inspeção no momento em que seria “esperado”, juntamente com a aeronave.

A principal implicação dessa característica é que para ser viável, uma rede de precedência de 1000 tarefas precisaria ser padronizada e replicável, e essa característica faz com que não se possa padronizar essa rede, pois uma grande quantidade de seus nós precisariam ser modificados para cada fora de fase existente no plano de manutenção. A tarefa de encontrar e modificar esses nós seria complexa dentro de uma rede tão grande, e aconteceria frequentemente. Todo esse trabalho não seria replicável, pois os fora de fase de uma aeronave são diferentes dos fora de fase de uma outra aeronave.

Isso não significa que a precedência não exista, no entanto ela está mais ligada a aspectos técnicos relacionados ao avião. Por exemplo, dois sistemas A e B não podem ser testados em paralelo se os sistemas A e B são operados da cabine de comando, necessitando assim que A seja testado primeiro e depois B, ou vice-versa. Outrossim é que esses sistemas A e B podem demandar a aeronave, por exemplo, energizada, enquanto podem haver tarefas de manutenção nos sistemas A e B, ou em outros sistemas, que requeiram o desligamento completo da aeronave. É também bastante comum que esses mesmos sistemas precisem ser desmontados para inspeções e troca de componentes.

Assim sendo, é adotada uma sequência padrão de grupos de atividades a serem realizadas na aeronave de forma a garantir respeito a essas principais precedências que precisam ser seguidas, com potencial de gerarem um alto impacto no *leadtime* da parada de manutenção. Essa sequência se repete para toda parada de manutenção, sendo indiferente o tamanho da manutenção programada. A Figura 2.2 abaixo ilustra a sequência básica de atividades que todo *check* de manutenção deve seguir.

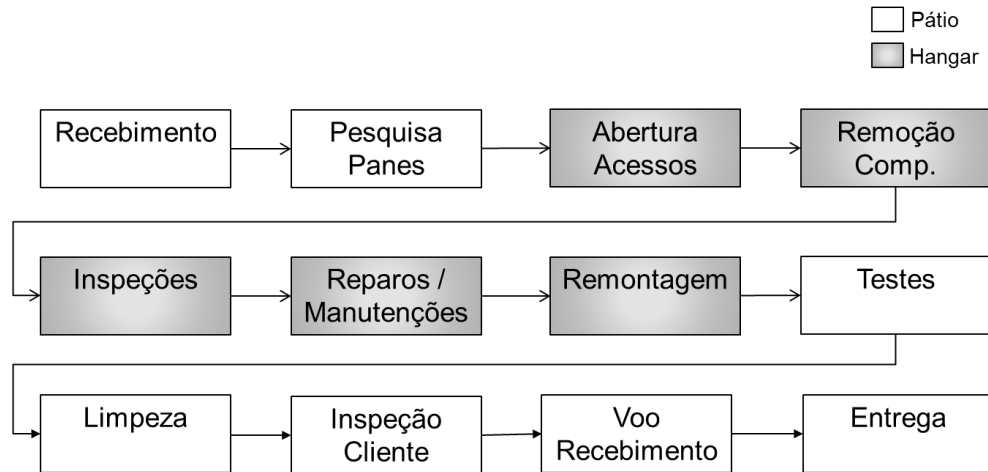


Figura 2.2: Sequência lógica básica de atividades de manutenção.

A Figura 2.2 ilustra as etapas lógicas que uma manutenção deve seguir. A aeronave deve ser recebida no centro de manutenção, ela fica no pátio onde é permitido o giro de motores, e então os problemas existentes podem ser pesquisados de forma que se determine o que precisa ser feito para sua correção. Na sequência, a aeronave pode ser colocada dentro do hangar, onde ela então é desmontada de maneira que se possa acessar os sistemas e componentes que fazem parte dessa manutenção. Os componentes que precisam ser reparados ou trocados são removidos e as inspeções são iniciadas. A seguir, os procedimentos dos reparos necessários são executados, para que então a aeronave possa ser remontada e testada. Por fim, o cliente inspeciona a aeronave e pode executar um voo para testes, para na sequência receber a aeronave de volta em definitivo. Toda parada de manutenção segue esta sequência, necessariamente, sob pena de caso essa sequência não seja respeitada, um problema que poderia ser evitado apareça apenas na fase de entrega ao cliente, gerando atrasos na entrega e consequentes transtornos junto ao mesmo.

Para que essa sequência seja viabilizada, as tarefas de manutenção são agrupadas de acordo com essas etapas, gerando pacotes de trabalho menores que devem ser executados de acordo com elas. Um outro agrupamento de tarefas é feito, dessa vez de acordo com os sistemas da aeronave dentro de cada pacote das etapas, o que deixará uma divisão das tarefas por etapa e sistemas, reduzindo novamente a pacotes de atividades ainda menores. Esses pacotes são entregues a times de técnicos de manutenção que serão responsáveis pela execução dos mesmos.

Nessa fase pouca precedência existe, em geral o número de tarefas é grande e o número de técnicos é limitado, fazendo com que a restrição seja o número de técnicos

disponíveis. O que poderia gerar um problema aqui seria espaço físico para execução das atividades, pois duas atividades que precisem ser executadas em um mesmo local não poderiam ser feitas ao mesmo tempo. No entanto, essa restrição é facilmente contornada por meio da distribuição das tarefas entre os técnicos em locais diferentes da aeronave. Isso pode ser facilmente executado, já que a aeronave é grande suficiente para tal. Outrossim, o controle de manutenção, o equivalente do controle de produção na indústria de transformação, busca ainda deixar tarefas nas mesmas regiões com os mesmos profissionais.

Uma vez definida a sequência de atividades a ser seguida, cabe explorar a aplicação de recursos. Na manutenção, assim como em toda Indústria da Aviação, os recursos são caros. Destacam-se três recursos principais: a mão de obra, os equipamentos de testes, e o hangar. Estes são os principais recursos utilizados e cuja falta pode impactar todo o projeto de manutenção.

Os hangares são as edificações onde as manutenções devem ocorrer. São caros, pois precisam ter características que encarecem suas edificações. Precisam ser grandes devido ao tamanho das aeronaves, e dependendo da quantidade de aeronaves que poderá estar em manutenção ao mesmo tempo, precisam ser grandes o suficiente para comportar mais de uma aeronave. Precisam ser altos, outra característica que encarece a edificação, pois o estabilizador vertical (Figuras 2.3 e 2.4) das aeronaves fica a alturas consideráveis, mesmo nas aeronaves de pequeno porte. Existem ainda atividades dentro dos planos de manutenção das aeronaves que devem ser feitas com a mesma, protegida das intempéries, pois a ela pode estar com diversos acessos abertos, expondo componentes que não devem ficar expostos à chuva e poeira, entre outros. Isso faz com que a programação da utilização do hangar seja relevante para os centros de manutenção. As Figuras 2.3 e 2.4 ilustram um estabilizador vertical de uma aeronave. A Figura 2.5 ilustra um hangar de manutenção.

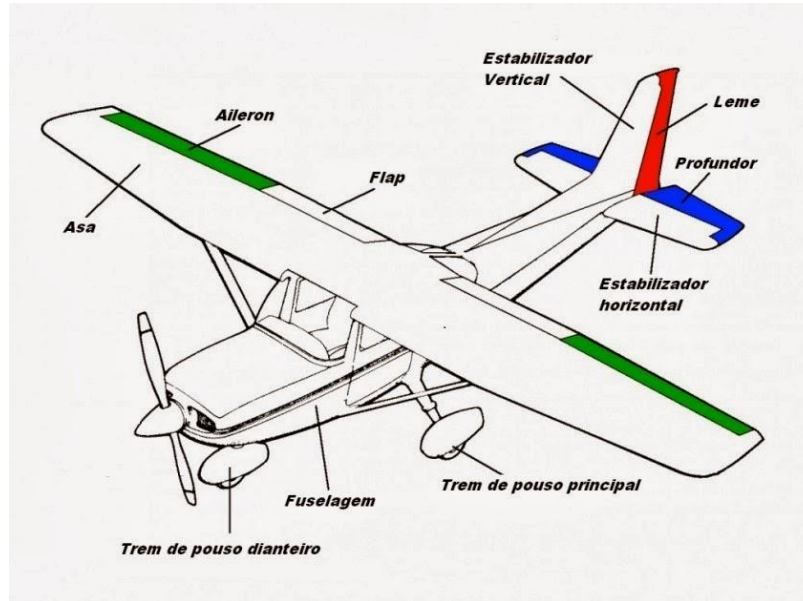


Figura 2.3: Ilustração de uma aeronave de pequeno porte (voetla.org).



Figura 2.4: Cauda de um Boeing 747 (wikipedia.org).



Figura 2.5: Hangar de manutenção Lufthansa Technik em Frankfurt (airwaysmag.com).

Neste trabalho, no entanto, não figura entre seus objetivos analisar a alocação do recurso hangar, mas certamente esse recurso seria digno de um trabalho interessante de pesquisa futura, dedicado na busca de otimizar a alocação de um recurso tão importante no universo da manutenção, principalmente quando o fluxo de aeronaves é alto.

As ferramentas de teste são também recursos importantes e caros, tais como bancadas de pressurização do sistema hidráulico, bancos de teste do sistema anemométrico, equipamentos de teste de sinais de rádio e balanças, entre outros. São equipamentos em geral de disponibilidade restrita em qualquer centro de manutenção, pois o tempo de utilização destes é pequeno diante do tempo total do projeto de manutenção como um todo, não justificando um investimento em diversas unidades desses equipamentos. Reside aí mais um problema de programação a ser resolvido, mas de custo reduzido se comparado ao custo do hangar de manutenção. Este trabalho também não estudará esses recursos, já que no centro de manutenção em estudo, os mesmos são facilmente compartilhados entre as aeronaves no dia a dia das atividades, sem gerar impactos significativos. Entende-se que a sua disponibilidade não é um grande problema.

A mão de obra é outro recurso da manutenção de relevante importância. É um recurso de baixa disponibilidade no mercado de trabalho, tem custo significativo, e a sua formação é lenta e burocrática. Um técnico de manutenção deve passar por uma formação teórica para depois poder trabalhar como auxiliar de manutenção. Deve permanecer por um ano e meio no mínimo nesta função (RBAC 65, 2018), para poder ser certificado como técnico de

manutenção aeronáutica pela ANAC. Apenas depois dessas etapas, o técnico estará apto a assinar e se responsabilizar pelo seu trabalho.

Durante o período como auxiliar, todas as suas atividades na aeronave deverão ser supervisionadas por técnico formado, que se responsabilizará por todas essas atividades. Toda esta burocracia, que é necessária e importante na formação da mão de obra, faz com que o processo de formação consuma tempo significativo, durante o qual a empresa permanece investindo no profissional.

Conforme já anteriormente mencionado, a mão de obra segue as especialidades relacionadas aos tipos de trabalhos a serem executados. Ao se analisar a MOD do local de estudo, nota-se que existem técnicos multidisciplinares, mas mesmo que existam alguns técnicos com qualificação em mais de uma especialidade, não o são em quantidade significativa de forma que mereça ser considerada nesse estudo.

2.3 MODELOS DE REFERÊNCIA

Conforme mencionado, o problema da manutenção de aeronaves pode ser abordado como um problema de gerenciamento de projetos. Pelas suas características, operações de projeto envolvem atividades complexas, geralmente de grande escala e com início e fim bem definidos (SLACK et al. 2009). Ainda segundo os autores, todo projeto tem uma meta definida e necessitará de um conjunto definido de recursos. O mais comum dos recursos em um projeto é o financeiro, e a maior parte dos projetos tem um custo financeiro associado. Muitas vezes o projeto possui uma meta de custo associada a ele.

Também conforme já mencionado, o problema aqui tratado pode ser visto como de programação de produção. Como tal, esse problema pode estar sujeito a restrições. A primeira delas, e comum no gerenciamento de projetos, é a restrição de precedência, que está preocupada com limitações na sequência em que certas atividades do projeto serão executadas. Ela tem importância relevante na programação de projetos, pois define se uma atividade pode ou não ser iniciada de acordo com suas predecessoras.

Segundo Conway et al. (1967), problemas de sequenciamento (*sequencing problems*) são comuns e na maioria das vezes são resolvidos casualmente ou automaticamente, sem que se tome conhecimento explícito da existência do problema. Para os autores, frequentemente as atividades são realizadas na sequência em que chegaram, pela regra do FIFO (*First In, First Out*), devido ao senso de “justiça” existente nessa regra. Segundo os autores, o

problema de sequenciamento de atividades não precisa ser resolvido assim no chão de fábrica, podendo-se adotar regras mais coerentes com os objetivos da empresa nesses casos.

Tanto Conway et al. (1967) quanto Baker (1974) definem o problema de sequenciamento (*sequencing*) puro como sendo o problema de sequenciar diversas atividades em uma única máquina, com um determinado objetivo. Segundo Conway et al. (1967), o problema de sequenciamento sincronizado de várias máquinas ao mesmo tempo passa a ser um problema de programação (*scheduling problem*). Ainda segundo estes últimos, essa é a máxima distinção que se pode fazer entre *sequencing* e *scheduling*, o que permite que esses termos possam ser tratados como sinônimos. No presente trabalho, tais termos também são considerados equivalentes.

Baker (1974), define *scheduling* como sendo a alocação de recursos no tempo para a execução de várias tarefas. Para Leung (2004), é a alocação de recursos escassos para execução de atividades com o objetivo de otimizar uma ou mais medidas de performance. Para Morton e Pentico (1993), *scheduling* é a decisão de coincidir atividades e recursos de forma a concluir trabalhos e projetos que necessitem dessas atividades oportunamente e com qualidade, ao mesmo tempo que se busca maximizar a taxa de saída, minimizando os custos de operação. Pinedo (2008), afirma que *scheduling* é a decisão de como lidar com a alocação de recursos às tarefas para um dado período de tempo, de maneira a otimizar um ou mais objetivos.

Outra restrição a que geralmente muitos projetos estão sujeitos é a de recursos. Exemplos de recursos podem ser: recursos financeiros, mão de obra, máquinas e equipamentos, entre outros. Este trabalho, conforme já explicado, se debruçará apenas sobre a análise do recurso mão de obra, o mais relevante no contexto deste estudo. Cada atividade do projeto demandará uma quantidade pré-determinada de um ou mais recursos para sua conclusão. Isso é relevante ao projeto, pois a quantidade disponível de certo recurso influenciará, por exemplo, a execução ou não de várias atividades que demandem esse recurso ao mesmo tempo.

Na presença da restrição de recursos têm-se então os *Resource Constrained Project Scheduling Problem* (RCPSP). Herroelen et al. (1998) definem esse problema como sendo o sequenciamento das atividades de um projeto de maneira a minimizar sua duração, sujeito a restrições de precedência entre as atividades, bem como a restrições de recursos no conjunto de recursos necessários. Segundo Pinedo (2005), o referido RCPSP na literatura é o problema de sequenciamento de projetos com restrição na força de trabalho (mão de obra). Para este autor, a força de trabalho pode consistir de uma série de grupos de operadores, e cada um pode ter especialidades diferentes.

Elmaghraby e Herroelen (1980) conjecturam que se os recursos estiverem disponíveis apenas em quantidades muito pequenas ou muito grandes, a complexidade de solução será baixa. No primeiro caso, haverá pouca liberdade para sequenciar as atividades, fazendo com que o projeto tenda a ser todo executado com estas em série, e a solução tenda a ser igual à soma da duração das atividades. No segundo caso, a solução tende a ter todas as atividades em paralelo, e a solução tende a ser igual ao caminho crítico. Pinedo (2005) afirma que o PSP (*Project Scheduling Problem*) básico, ou seja, sem a restrição da mão de obra, é fácil de ser resolvido computacionalmente e soluções ótimas podem ser obtidas com pouco esforço computacional. O problema com restrição de mão de obra, por sua vez, é muito difícil de ser resolvido otimamente em determinadas situações, dependendo do tamanho da instância. Herroelen (2005) afirma que o RCPSP é classificado como NP-difícil, uma vez que o mesmo é uma generalização do problema de sequenciamento de um *job shop*.

Pritsker et al. (1968) propõe um dos primeiros modelos de programação de projetos com restrições de recursos. Ele utiliza uma formulação de programação linear com variáveis binárias. Seu modelo foi um diferencial na pesquisa de programação de projetos na época, pois foi capaz de lidar com diversas restrições ao mesmo tempo, adequando-se a diversas aplicações reais. Considerado o principal precursor dos modelos exatos clássicos hoje utilizados nos problemas RCPSP, a formulação de Pritsker et al. (1968) é para múltiplos projetos, e em seu trabalho eles formulam o modelo para três funções objetivo diferentes: Minimizar o tempo total de atravessamento; Minimizar o *makespan*; e Minimizar o atraso total (ou a penalidade pelo atraso). Na sequência, apresentam as principais restrições, sendo que estas são todas até hoje consideradas em diversos trabalhos.

Os autores apresentam uma restrição de conclusão de tarefa (*job completion*), que define que cada *job* terá exatamente um período de conclusão. Uma restrição de conclusão do projeto (*project completion*), e define que todo o projeto só terá sido concluído quando todas as atividades forem realizadas. São apresentadas as restrições de precedência e de recursos. A restrição de precedência é definida limitando que o período de término da tarefa predecessora, somado à sua duração, deve ser menor que o instante de término da tarefa sucessora. Já a restrição de recursos, define que a somatória dos recursos utilizados nas tarefas deve ser menor que a quantidade do recurso do tipo disponível.

Pritsker et al. (1968) trabalha ainda com diversas outras restrições. Restrição de substituição dos recursos para situações em que, por exemplo, uma pessoa altamente qualificada pode ser substituída por outra menos qualificada em tarefas específicas. Restrição de simultaneidade, ou seja, quando tarefas diferentes devem ser feitas simultaneamente. Restrição

de divisão do trabalho (*job splitting*), ou seja, em que uma tarefa pode ser interrompida antes de seu término para que outra seja iniciada. No caso da divisão do trabalho, o autor considera ainda situações de custos adicionais em decorrência dessa divisão, ou ainda o aumento da duração da tarefa como resultado da mesma.

O *job splitting* mencionado por Pritsker et al. (1968) é bastante abordado na literatura e é, atualmente, mais citado como preempção (*preemption*). Segundo Pinedo (2005) preempção ocorre quando o programador precisa interromper uma tarefa que esteja em execução, por exemplo, em uma máquina, para que a mesma fique disponível para uma outra tarefa. Diz-se da tarefa paralisada que a mesma é uma tarefa interrompida. Segundo Herroelen et al. (1998) e Hartmann e Briskorn (2009), no modelo RCPSP básico, uma vez iniciada uma tarefa, a mesma não pode ser interrompida, ou seja, não é permitido preempção.

Pinedo (2005) apresenta uma formulação inteira do modelo RCPSP que tem sido bastante utilizada na literatura. Os tempos de processamento são determinísticos e inteiros. Essa formulação é mais atual e pode ser considerada uma evolução do trabalho de Pritsker et al. (1968), podendo ser mais facilmente relacionada ao modelo proposto neste trabalho.

Resource Constrained Project Scheduling Problem

Índices:

- j (ou k) : tarefa, $j = \{1, \dots, n\}$.
 t (ou u) : período, $t = \{1, \dots, H\}$.
 l : tipo de mão de obra, $l = \{1, \dots, Np\}$.

Parâmetros:

- $n + 1$: tarefa fictícia, de duração zero, que sucede todas as demais.
 p_k : períodos de duração da tarefa j , $1 \leq p_j \leq T$, e inteiro.
 W_{lj} : quantidade de mão de obra do tipo l utilizada na tarefa j .
 W_l : quantidade de mão de obra do tipo l disponível.
 H : limitante superior do *makespan*.
 $A = \{(j,k) \mid \text{a tarefa } j \text{ precede a tarefa } k\}$.

Variáveis:

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

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

Sujeito à:

$$\sum_{t=1}^H t x_{jt} + p_k - \sum_{t=1}^H t x_{kt} \leq 0 \quad \text{para } (j, k) \in A \quad (2)$$

$$\sum_{j=1}^n \left(W_{lj} \sum_{u=t}^{t+p_j-1} x_{ju} \right) \leq W_l \quad \begin{array}{l} \text{para } l = \{1, \dots, N_p\} \\ \text{para } t = \{1, \dots, H\} \end{array} \quad (3)$$

$$\sum_{t=1}^H x_{jt} = 1 \quad \text{para } j = \{1, \dots, n\} \quad (4)$$

$$x_{jt} \in \{0,1\} \quad \text{para } \forall j, \forall t \quad (5)$$

A função objetivo (1) busca minimizar a *makespan*. A restrição (2) tem a função de garantir respeito às precedências entre as tarefas. As restrições (3) garantem respeito às quantidades disponíveis dos recursos do tipo l . A restrição (4) garante que todas as tarefas sejam programadas. Por fim, a restrição (5) indica o domínio das variáveis.

Silva (2010) e Silva et al. (2011) propõem um modelo que aborda o sequenciamento de atividades de produção e montagem em gabaritos, com restrições de adjacência na Indústria Aeronáutica, problema semelhante ao que ocorre quando existem problemas de espaço na manutenção de aviões. Em um primeiro momento, acreditou-se que o presente trabalho deveria seguir em direção semelhante. À medida que se aprofundou no problema, notou-se que o espaço se torna um problema real em aeronaves menores, no entanto, as aeronaves da frota, objeto deste estudo, possuem uma dimensão grande o suficiente para que essa restrição não precise ser levada em consideração. Por esse motivo, será considerado aqui que as tarefas dentro dos pacotes das etapas podem ser executadas em paralelo.

Silva et al. (2011) tratam o problema de montagem em gabaritos como um problema de programação de máquinas paralelas com restrições de adjacência e sequência, propondo um modelo de programação linear inteira mista (MILP – *Mixed Integer Linear Programming*). O trabalho testa o modelo proposto fazendo uso de problemas fictícios para validação, sendo que os resultados são satisfatórios e o modelo gera soluções factíveis. O trabalho realiza ainda testes com exemplos reais e mostra que as soluções obtidas, resolvendo-se o modelo, são melhores que as inicialmente utilizadas na situação real. Em um segundo momento, o estudo testa dois tipos de programação das atividades para as situações reais: programação por peças, de acordo com uma demanda fornecida para um determinado intervalo de tempo no qual todas as peças devem ser entregues; e programação por aeronave, na qual o modelo é utilizado para encontrar uma programação que entregue todas as peças do conjunto em um intervalo de tempo que atenda à demanda do conjunto no tempo.

É possível enxergar similaridade do problema de Silva (2010) e Silva et al. (2011) com o problema de sequenciamento das atividades de manutenção em aeronaves, em que se poderia sequenciar as atividades de manutenção buscando reduzir ou minimizar o tempo necessário para a entrega de todas as tarefas, ou então se poderia buscar sequenciar diversas aeronaves a partir de uma solução padrão previamente estudada. Em sua conclusão, os autores avaliam que a programação por peças tende a gerar uma variação do tempo de produção de itens iguais, gerando uma incerteza no suprimento de montagens subsequentes que demandarão estes itens. O trabalho ressalta ainda que a menor variabilidade da solução obtida com a programação por aeronave permite mais facilidades de implantação prática, menor estoque em processo e menor *leadtime* de produção, levando à melhor utilização dos recursos e menos horas extras.

Guldmond et al. (2008) propõem uma abordagem para o sequenciamento de atividades com restrições de tempo, aplicando-a para o TCPSP (*Time Constrained Project Scheduling Problem*), ou problema de sequenciamento de atividades de projetos com restrição de tempo. Segundo os autores, esse problema é na verdade uma variação do RCPSP (*Resource Constrained Project Scheduling Problem*), ou problema de sequenciamento de atividades com restrição de recursos. As principais diferenças entre esses dois problemas residem em que no TCPSP os prazos de entrega são estritos e não modificáveis e a capacidade dos recursos pode ser modificada de maneira a que se atendam esses prazos, enquanto que no RCPSP as quantidades de recursos disponíveis não são variáveis e não podem ser ultrapassadas, sendo que o objetivo principal é minimizar o *makespan*.

Na gestão da maioria dos projetos, sabe-se que prazos são fortes restrições, e ainda segundo Guldmond et al. (2008), existem diferentes maneiras de acelerar as atividades de forma a cumprir os prazos estabelecidos dos projetos, como utilizar horas extras ou contratar capacidade adicional. A questão principal é qual tipo de capacidade adicional utilizar, o quanto contratar e quando contratar, de maneira a minimizar os custos. O trabalho ressalta ainda que embora o problema restrito no tempo seja comum, no cotidiano da gestão de projetos, ele ainda é pouco explorado.

Na realidade de um centro de manutenção, variar a quantidade MOD disponível não é algo trivial. Conforme já mencionado, a disponibilidade desses profissionais no mercado não é grande, sua formação é lenta e a regulamentação do setor aeronáutico impõe restrições à contratação de MOD pouco qualificada. De fato, nenhum centro de manutenção se dispõe a usar MOD de baixa qualificação, a exemplo do que pode ser feito em outros tipos de indústrias, dado o risco à que se exporia a segurança dos tripulantes e dos passageiros. Fica ao final a opção de utilização de horas extras, o que mediante a legislação trabalhista brasileira, que limita a jornada de trabalho, acaba sendo uma opção bastante utilizada mas bastante restrita, gerando um número de horas homem adicionais não muito significativo.

Nas manutenções em estudo nesse trabalho, o tempo de término das mesmas é restrito, importante e contratualmente limitado, podendo levar a muitas caso os tempos contratados para a liberação das aeronaves não sejam cumpridos. Dadas as dificuldades e restrições em se variar a aplicação adicional de MOD, o que se pretende é, considerando as restrições de MOD, avaliar se um modelo RCPSP é capaz de melhorar o prazo de entrega das tarefas programadas e, conseqüentemente, das aeronaves.

Silva et al. (2014a, 2014b) propõem um modelo de programação linear inteira com o objetivo de minimizar a necessidade de aplicação de MOD em uma montagem aeronáutica composta por duas etapas, uma em gabarito e uma em bancada, ainda levando em consideração uma restrição de adjacência, que impede o uso das posições adjacentes do gabarito no momento em que uma tarefa está sendo feita, por questão de espaço físico. O estudo considera quatro fases distintas, relacionadas à curva de aprendizagem da mão de obra, em que da fase 1 para a fase 4, a equipe atinge o máximo de especialização. Nessa última fase, todos os operadores são considerados capazes de realizar todas as tarefas, ou seja, existe apenas um tipo de mão de obra.

Na manutenção de aeronaves, esse nível de especialização é mais difícil de ser atingido, já que o que proporciona a especialização é a repetição das atividades. Muito embora exista alguma repetição em toda atividade, como as atividades de torque e freio em um parafuso

ou a inspeção de uma estrutura, a manutenção de aeronaves nunca viverá a repetição de uma linha de montagem, onde uma tarefa pode chegar a ser repetida diariamente, ou até mesmo diversas vezes no mesmo dia. É comum em uma linha de manutenção a realização de atividades com manual de manutenção em mãos e a realização de cada uma das etapas com o auxílio deste.

Na fase dois do trabalho, Silva et al. (2014a, 2014b) propõem um modelo TCPSP que considera a restrição de mão de obra e restrição de recurso. O modelo minimiza a aplicação de MOD, e com a restrição de recurso do modelo são abordadas as limitações do gabarito e a restrição de MOD. Esta última não permite a utilização deste recurso além do disponível. Os autores concluem que a abordagem proposta consegue resolver o problema em tempo computacional razoável para a realidade de uma empresa, com ganho de produtividade. Um resultado interessante foi a composição de um perfil de quantidade de MOD ideal, necessária para realizar as montagens para cada cadência de produção possível, o que permite ao gerente do processo produtivo tomar a melhor decisão ao precisar variar os níveis de produção.

Silva e Morabito (2015) usam os modelos de programação de projetos explorando uma montagem de grandes segmentos aeronáuticos, objetivando avaliar o impacto do *leadtime* sobre a quantidade e o nivelamento da mão de obra. Os autores avaliam a utilização de múltiplos modos de execução, em que as mesmas atividades podem ser feitas em menos tempo com mais mão de obra. O estudo utiliza inicialmente um modelo clássico RCPSP para avaliar a relação entre *leadtime* da montagem e mão de obra, e um modelo MMRCPSPP (*Multi Mode Resource Constrained Project Scheduling Problem*) para estudar os múltiplos modos de execução.

Os autores resolvem o problema RCPSP muitas vezes, variando o *leadtime* para execução do projeto e chegando a conclusões importantes. A primeira conclusão é que ao aumentar o *leadtime*, a quantidade de mão de obra necessária diminui, pois ao fazer isso, a gama de soluções possíveis aumenta, permitindo soluções melhores do ponto de vista da utilização da MOD. Após certo limite do *leadtime* de montagem, a MOD pára de diminuir, tendendo ao equilíbrio. A segunda conclusão dos autores é que juntamente com a redução, o nivelamento da MOD também melhora e, conseqüentemente, a utilização dos operadores aumenta. Por fim, os autores concluem que a utilização dos múltiplos modos de execução é mais vantajosa em relação à de único modo, pois encontra soluções melhores e com menos utilização de MOD para os casos em que os dois modelos encontraram soluções para os mesmos problemas. Adicionalmente, o modelo de múltiplos modos encontrou soluções factíveis em uma grande parte dos casos em que o modelo de único modo não encontrou nenhuma solução.

Yamashita e Morabito (2007, 2009) propõe um algoritmo exato de programação de projetos, que considera múltiplos modos de execução e os custos relacionados a disponibilidade da mão de obra. O algoritmo é baseado na solução do problema MMRACP (*Multi Mode Resource Availability Cost Problem*), e busca soluções viáveis para ele variando a disponibilidade do recurso (múltiplos modos). O algoritmo gera curvas de *trade-off* entre o custo e o prazo do projeto. O aspecto relevante é que tais curvas podem facilitar a análise do tomador de decisão. Os autores avaliam que o algoritmo é computacionalmente viável para problemas de tamanho moderado, com os recursos disponíveis considerados em quantidades pequenas.

Embora seja uma possibilidade bastante interessante para o centro de manutenção, a utilização de múltiplos modos não será abordada nesse trabalho, pois o objetivo principal neste momento é avaliar as possibilidades de ganhos com um único modo de execução.

Conforme comentado no Capítulo 1, o presente trabalho pretende propor um modelo baseado em abordagens de programação de projetos para a manutenção de aeronaves. Pretende-se usar um modelo RCPS, com o objetivo de minimizar o *makespan* e o *leadtime* médio das atividades planejadas para um determinado horizonte de tempo conhecido.

Neste trabalho, é utilizado como base o modelo proposto por Silva et al. (2014b) na fase 2 do trabalho, oriundo dos trabalhos clássicos de Pritsker et al. (1968) e Pinedo (2005). Por esse motivo ele é aqui apresentado:

Índices:

- j (ou h): tarefa, $j = \{1, \dots, J\}$.
 t (ou b): período, $t = \{1, \dots, T\}$.
 k : tipo de recurso, $k = \{1, \dots, K\}$.
 w : tipo de mão de obra, $w = \{1, \dots, W\}$.

Parâmetros:

- r_j : período de liberação da tarefa j , $1 \leq r_j \leq T$, e inteiro.
 d_j : período conclusão da tarefa j , $1 \leq d_j \leq T$, e inteiro.
 rec_{jk} : quantidade de recurso k necessárias para executar a tarefa j
 c_k : quantidade disponível do recurso k .
 p_j : períodos de duração da tarefa j .

n_{jw} :	quantidade de mão de obra do tipo w utilizada na tarefa j .
v_w :	custo unitário da mão de obra do tipo w .
H :	$\{(h,j) \mid \text{a tarefa } h \text{ precede a tarefa } j\}$.
G :	$\{j \mid \text{a tarefa } j \text{ é executada no gabarito}\}$.

Variáveis:

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

a_w : quantidade de mão de obra do tipo w disponível.

$$\text{Min} \sum_{w=1}^W v_w \cdot a_w \quad (6)$$

$$\sum_{t=1}^T x_{jt} = 1 \quad \text{para } j = \{1, \dots, J\} \quad (7)$$

$$\sum_{t=r_h+p_h}^{d_h} t \cdot x_{ht} \leq \sum_{t=r_j+p_j}^{d_j} (t - p_j) \cdot x_{jt} \quad \text{para } \forall (h,j) \in H \quad (8)$$

$$\sum_{j \in G} \sum_{b=t}^{t+p_j-1} rec_{jk} \cdot x_{jb} \leq c_k \quad \text{para } \forall (k,t) \quad (9)$$

$$\sum_{j=1}^J \sum_{b=t}^{t+p_j-1} n_{jw} \cdot x_{jb} \leq a_w \quad \text{para } \forall (w,t) \quad (10)$$

$$x_{jt} = 0 \quad \begin{array}{l} t = 1, \dots, r_j + p_j - 1 \\ t = d_j + 1, \dots, T \\ \text{para } j = \{1, \dots, J\} \end{array} \quad (11)$$

$$x_{jt} \in \{0,1\} \quad \text{para } \forall j, \forall t \quad (12)$$

$$a_w \in \mathbb{Z}^+ \quad \text{para } \forall w$$

A função objetivo (6) minimiza o custo da mão de obra. A equação (7) garante que cada tarefa seja alocada uma única vez. A inequação (8) garante respeito às restrições de precedência entre as tarefas. A inequação (9) limita a quantidade de recursos disponíveis,

garantindo a restrição de adjacência proposta pelos autores. A inequação (10) limita a quantidade de mão e obra do tipo w disponível. As restrições (11) garantem respeito às datas de entrega e chegada das tarefas. Por fim as restrições (12) tratam do domínio das variáveis de decisão.

Cabe ressaltar que embora os autores do trabalho tenham proposto um modelo TCPSP, o modelo proposto no presente trabalho é um RCPSP, já que os recursos são restritos e o objetivo é minimizar o tempo de realização de todas as tarefas programadas. Isso não é uma dificuldade, pois conforme já abordado, o problema estrito no tempo TCPSP, é um caso particular do RCPSP. Para tal, o modelo de referência terá a função objetivo alterada, para contemplar a função objetivo deste trabalho, e as restrições de adjacência (9) não serão utilizadas. O modelo já adaptado para este trabalho é apresentado em detalhes no Capítulo 3.

3 FORMULAÇÃO MATEMÁTICA DO PROBLEMA

As atividades de manutenção seguem uma sequência pré-definida por uma lógica que considera aspectos técnicos, alguns até certo ponto óbvios, como realizar determinadas atividades que demandam a aeronave montada antes das atividades de desmontagem.

Na Figura 3.1 é mostrada a rede de precedência propriamente dita de uma manutenção de aeronave realizada no centro de manutenção objeto deste trabalho. Essa rede possui dezesseis grupos de atividades, que precisam ser realizados de forma a concluir toda a manutenção. A Figura 2.2 ilustra a sequência básica de como deve ser organizado o processo de manutenção, sequência esta que provavelmente deve ser adotada por outros centros de manutenção.

A rede da Figura 3.1 traduz essa lógica de maneira prática e de acordo com complexidades técnicas adicionais existentes na realidade da manutenção. Por isso, conforme é possível notar pela figura da rede, sua complexidade é maior que na Figura 2.2. Essa rede não será detalhada de forma a explicar cada um de seus nós, a fim de resguardar os processos da empresa, pois isso mostraria a forma como a empresa organiza suas atividades de manutenção, tecnicamente. Aqui basta saber que essa rede respeita a lógica da Figura 2.2, mas contempla outras questões técnicas relevantes no processo de manutenção.

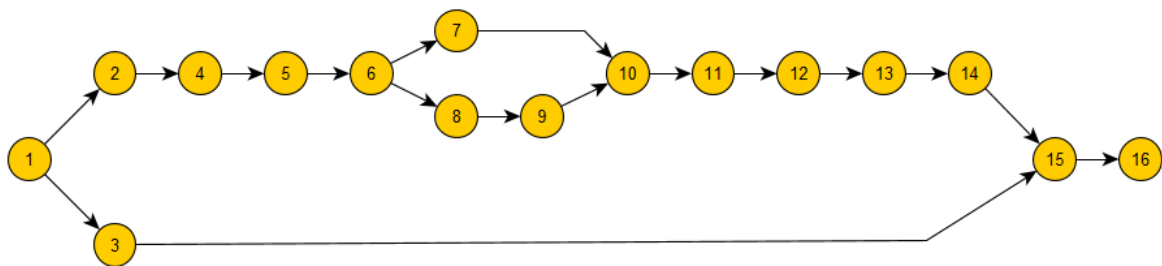


Figura 3.1: Rede de precedência do processo de manutenção.

Para execução da manutenção, a área de planejamento agrupa todas as tarefas de manutenção de acordo com a rede da Figura 3.1. Dessa forma, deve-se entender que cada tarefa dessa rede representa não apenas uma única atividade, mas sim um pacote de atividades que deve ser completamente realizado, antes que o próximo nó da rede seja iniciado. Para todas as

tarefas existentes, o empacotamento de acordo com essa rede já está previamente cadastrado no sistema da empresa.

É provável que outros centros de manutenção sigam lógicas semelhantes, customizando-as de acordo com os modelos de aeronave a que atendem, ou ainda de acordo com a filosofia de manutenção adotada, expertise técnico, entre outros. Ressalta-se, entretanto, que em uma parada de manutenção, muito embora existam manutenções bastante simples, raramente terá apenas dezesseis atividades. Para a família de aeronaves estudada, as menores paradas de manutenção possuem aproximadamente sessenta tarefas e as maiores podem chegar a mil e duzentas atividades. Para as menores paradas, as aeronaves ficam indisponíveis por apenas alguns dias, enquanto que nas maiores, esse tempo pode chegar a um mês e mais alguns dias.

Uma vez compreendido que os nós da rede representam não apenas uma atividade, pode existir precedência entre as tarefas dentro dos pacotes de cada nó. No dia a dia da operação, essas tarefas são mapeadas e distribuídas pelo controle de manutenção aos técnicos, respeitando essas precedências. Em geral, dentro de cada nó existem muitas tarefas e são poucas as que possuem precedência a ser seguida. Para lidar com as tarefas que possuem restrição de precedência, o controle da manutenção distribui essas tarefas de forma que elas sejam concluídas antes que todo o pacote seja terminado. Cabe ressaltar que isso é possível, pois existe, sempre, um volume de atividades que permite sequenciá-las de forma que aquelas com precedência não estejam no caminho crítico.

No presente trabalho, por simplificação, o problema de estudo é trabalhado da mesma forma, considerando que cada nó da rede de precedência é uma atividade, de *leadtime* igual ao praticado pela empresa, quando feito o planejamento do pacote de manutenção. Isso é feito, pois, se for levado em consideração o problema puro, este poderia chegar a um tamanho muito grande, ao partir de um número elevado de tarefas combinado ao horizonte de tempo que precisaria ser analisado. Dessa forma, a abordagem adotada permitirá analisar qualquer tamanho de problema.

O centro de manutenção, objeto deste estudo, é capaz de trabalhar em diversas aeronaves ao mesmo tempo, sendo que no hangar principal existem quatro posições de aeronaves que podem ficar dentro dele ao mesmo tempo. Esse centro possui ainda mais quatro posições de aeronaves no pátio de serviço. O que se pretende é buscar uma solução para o sequenciamento das atividades de manutenção em mais de uma aeronave ao mesmo tempo, buscando otimizar o tempo gasto nessas manutenções.

São apresentados a seguir dois modelos de programação de atividades de manutenção de aeronaves, ambos propostos neste trabalho. Os modelos são iguais em todas as restrições, diferenciando-se apenas em suas funções objetivo. A formulação é baseada na estrutura RCPSP, ou problema de programação de projetos com recursos restritos. Trata-se de um problema de programação inteira mista: o primeiro modelo com função objetivo de minimizar o *leadtime* médio de conclusão das atividades, e o segundo com o objetivo de minimizar o *makespan*. No primeiro caso isso é obtido ao minimizar o somatório do instante de término de todas as últimas tarefas de manutenção de cada aeronave, e no segundo ao minimizar a última atividade de todo o projeto.

A seguir é apresentado o modelo com função objetivo de minimizar *leadtime* médio:

Índices:

- j (ou h): tarefa, $j = \{1, \dots, J\}$.
 t (ou b): período, $t = \{1, \dots, T\}$.
 w : tipo de mão de obra, $w = \{1, \dots, W\}$.
 f_{im_v} : última tarefa de cada aeronave, $v = \{1, \dots, V\}$.
 FIM : última atividade do projeto (fictícia).

Parâmetros:

- r_j : período de liberação da tarefa j , $1 \leq r_j \leq T$, e inteiro.
 d_j : período de conclusão da tarefa j , $1 \leq d_j \leq T$, e inteiro.
 p_j : períodos de duração da tarefa j , $1 \leq p_j \leq T$, e inteiro.
 n_{jw} : quantidade de mão de obra do tipo w utilizada na tarefa j .
 a_w : quantidade de mão de obra do tipo w disponível.
 $H = \{(h,j) \mid \text{a tarefa } h \text{ precede a tarefa } j\}$.

Variáveis:

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

$$\text{Min} \sum_{t=1}^T \sum_{v=1}^V t \cdot x_{f_{im_v}, t} \quad (13a)$$

$$\sum_{t=1}^T x_{jt} = 1 \quad \text{para } \forall j \quad (14)$$

$$\sum_{t=r_h+p_h}^{d_h} t \cdot x_{ht} \leq \sum_{t=r_j+p_j}^{d_j} (t - p_j) \cdot x_{jt} \quad \text{para } \forall (h, j) \in H \quad (15)$$

$$\sum_{j=1}^J \sum_{b=t}^{t+p_j-1} n_{jw} \cdot x_{jb} \leq a_w \quad \text{para } \forall w, \forall t \quad (16)$$

$$x_{jt} = 0 \quad \begin{array}{l} t = 1, \dots, r_j + p_j - 1 \\ t = d_j + 1, \dots, T \\ \text{para } \forall j \end{array} \quad (17)$$

$$x_{jt} \in \{0,1\} \quad \text{para } \forall j, \forall t \quad (18)$$

A função objetivo (13a) busca minimizar a somatória dos instantes de término da última atividade de cada aeronave, ou seja, o numerador do *leadtime* médio na equação (19). A última atividade de cada aeronave é representada pelo índice fim_v , sendo que fim_1 é tarefa final da aeronave 1, fim_2 a tarefa final da aeronave 2, e fim_V a tarefa final da última aeronave V do projeto. Ao minimizar o somatório dessas atividades é obtida, portanto, a minimização do *leadtime* médio. Note que não é necessário dividir o numerador pela quantidade de aeronaves a exemplo da equação (19) abaixo, sem prejuízo do resultado, pois nessa formulação V é uma constante. Por essa razão, e por simplicidade matemática e computacional, optou-se por formular o modelo utilizando a função objetivo (13a) e não conforme a função objetivo (19) abaixo. O *leadtime* médio mínimo, pode ser obtido por um pós-processamento da função objetivo, conforme a expressão (19), mostrada a seguir.

$$Min \sum_{t=1}^T \sum_{v=1}^V \frac{t \cdot x_{fim_v,t}}{V} \quad (19)$$

A equação (14) garante que cada tarefa seja programada. A inequação (15) é a restrição de precedência e garante que todas as precedências existentes entre as tarefas sejam respeitadas. A inequação (16) garante respeito às quantidades de mão de obra de cada tipo w disponíveis. As restrições (17) garantem respeito aos períodos de chegada e término das tarefas. Por fim, a restrição (18) indica o domínio das variáveis.

Entre as principais diferenças entre os modelos (13a)-(18) deste capítulo e (6)-(12) do Capítulo 2, está a função objetivo. Em (6) a função objetivo busca minimizar o custo da mão de obra. Já em (13a) essa função minimiza o *leadtime* médio, tendo sido adaptada para minimizar o somatório dos instantes t em que termina a última atividade de cada aeronave. Esse instante de término de cada aeronave é representado por t_{fim} . Vale ressaltar que o *makespan* da solução desse modelo deve ser obtido como um pós-processamento da solução do modelo, observando os períodos t de término das atividades de cada avião e selecionando o valor máximo entre eles.

Outra diferença importante entre os modelos está nas restrições (9), (10) e (16). Vale notar que essas inequações são estruturalmente idênticas, e representam uma restrição de recurso. As restrições (10) e (16), são equivalentes nos dois modelos e representam restrição à quantidade do recurso mão de obra. A restrição (9) por sua vez, é utilizada naquele modelo para garantir a restrição de adjacência abordada em Silva et al. (2014b). Essa restrição de adjacência, conforme já mencionado, não possui equivalente no modelo (13a)-(18), ou no modelo (13b)-(18) adiante. O leitor que desejar melhor compreensão dessa restrição de adjacência pode encontrar detalhes em Silva et al. (2014b).

A seguir é apresentado o modelo (13b)-(18), cuja única diferença está na função objetivo. Para melhor efeito didático as restrições são novamente reproduzidas.

$$Min \sum_{t=1}^T t \cdot x_{FIM,t} \quad (13b)$$

$$\sum_{t=1}^T x_{jt} = 1 \quad para \forall j \quad (14)$$

$$\sum_{t=r_h+p_h}^{d_h} t \cdot x_{ht} \leq \sum_{t=r_j+p_j}^{d_j} (t - p_j) \cdot x_{jt} \quad para \forall (h,j) \in H \quad (15)$$

$$\sum_{j=1}^J \sum_{b=t}^{t+p_j-1} n_{jw} \cdot x_{jb} \leq a_w \quad para \forall w, \forall t \quad (16)$$

$$x_{jt} = 0 \quad \begin{array}{l} t = 1, \dots, r_j + p_j - 1 \\ t = d_j + 1, \dots, T \\ para \forall j \end{array} \quad (17)$$

$$x_{jt} \in \{0,1\} \quad \text{para } \forall j, \forall t \quad (18)$$

A função objetivo (13b), busca minimizar o *makespan* diretamente, sem necessidade de pós processamento como no modelo anterior. Vale destacar que *FIM* representa a última atividade da rede, uma atividade fictícia de tempo de processamento igual a zero, que sucede todas as últimas atividades de cada aeronave. Sendo assim, o índice *FIM* é diferente de *fim_v*. No Capítulo 4, tais diferenças ficarão mais claras com a apresentação das redes completas dos problemas abordados. As restrições (14)-(18) são as mesmas do modelo (13a)-(18) e já foram detalhadas.

Um aspecto importante é notar que no presente trabalho se optou pela formulação em tempo discreto. Formulações em tempo contínuo poderiam ter sido exploradas, mas isso traria complexidade adicional à formulação de algumas restrições, por exemplo a restrição (16), o que levou à escolha da discretização.

Por fim, um outro ponto merece uma reflexão na diferença entre as funções objetivo (13a) e (13b). Por sua característica construtiva, ao minimizar o somatório dos *leadtimes* de cada aeronave, ou seja o *leadtime* médio, espera-se que o modelo (13a)-(18) tenha uma tendência a compactar mais as atividades de cada aeronave individualmente. Opção que pode ser interessante para aplicação prática. Já a função objetivo (13b) tende a “enxergar” apenas o projeto como um todo, e assim, espera-se que o modelo (13b)-(18) minimize o projeto todo, mesmo que em detrimento das aeronaves individualmente.

4 EXPERIMENTOS COMPUTACIONAIS

Os modelos (13a)-(18) e (13b)-(18) do Capítulo 3 foram implementados na linguagem GAMS e resolvidos pelo *solver* CPLEX. A configuração do *hardware* utilizado para solução foi um computador com processador i7 de 3.40 GHz e 16 GB de memória RAM. Para todos os experimentos, foi utilizado o GAMS 24.3.1 juntamente com o CPLEX 12.5.1.0.

4.1 EXPERIMENTOS DE TESTE

Nessa fase foram realizados treze experimentos de teste com a finalidade de testar a corretude e implementação do modelo. O objetivo principal foi de avaliar se o modelo é capaz de gerar soluções factíveis respeitando as restrições dentro de limites de tempo aceitáveis. Outros objetivos foram verificar a formulação da função objetivo, a capacidade do modelo de encontrar soluções, e por fim, avaliar o crescimento da dificuldade computacional de resolver o problema frente ao aumento do tamanho do problema.

Foram estudados três tamanhos de problema, para uma aeronave, duas aeronaves e quatro aeronaves. Para todas as situações, a precedência considerada foi exatamente a rede de precedências da Figura 3.1. Para os parâmetros de duração das tarefas e quantidade de mão de obra necessária para sua execução, foram utilizados dados fictícios, propositalmente pequenos, diferentes de zero e inteiros, de forma que permitissem que se conhecesse a solução esperada a priori. Para a quantidade de mão de obra disponível, já que essa quantidade é um parâmetro do modelo, foram efetuados dois experimentos para cada um dos três tamanhos diferentes de problema, o que resultou em três pares de experimentos.

Foram ainda analisadas para cada tamanho de problema as duas funções objetivo que este trabalho pretende estudar, otimizar o *leadtime* médio e otimizar o *makespan*, o que elevou o número de experimentos de teste para seis pares. Os pares de experimentos de 1 a 3 foram realizados com a função objetivo do *leadtime* médio, e os pares 4 a 6 com a função objetivo do *makespan*. A Tabela 4.1 diferencia as principais características dos experimentos.

Tabela 4.1: Principais características dos experimentos de teste.

Teste	N.º Aeronaves	Função Objetivo	MOD	Teste	N.º Aeronaves	Função Objetivo	MOD
1.1	1	Leadtime Médio	$a_l = 5$, Abundante	4.1	1	Makespan	$a_l = 5$, Abundante
1.2	1	Leadtime Médio	$a_l = 1$, Restrita	4.2	1	Makespan	$a_l = 1$, Restrita
2.1	2	Leadtime Médio	$a_l = 5$, Abundante	5.1	2	Makespan	$a_l = 5$, Abundante
2.2	2	Leadtime Médio	$a_l = 1$, Restrita	5.2	2	Makespan	$a_l = 1$, Restrita
3.1	4	Leadtime Médio	$a_l = 5$, Abundante	6.1	4	Makespan	$a_l = 5$, Abundante
3.2	4	Leadtime Médio	$a_l = 1$, Restrita	6.2	4	Makespan	$a_l = 1$, Restrita
3.2b	4	Leadtime Médio	$a_l = 1$, Restrita	-	-	-	-

Nesta seção, a fim de comprovar seus objetivos, serão detalhados os resultados dos experimentos de 1.1 a 3.2, experimentos estes com a função de minimizar o *leadtime* médio e resumidos na Tabela 4.2 adiante. Os experimentos de 4.1 a 6.2 (minimizando o *makespan*) têm seus resultados apresentados ao fim desta seção, na Tabela 4.3. Os experimentos, 1.1 e 4.1, 2.1 e 5.1 e 3.1 e 6.1, são equivalentes em todos os seus parâmetros, diferenciando-se apenas pela função objetivo. Os experimentos, 1.2 e 4.2, 2.2 e 4.2 e 3.2 e 6.2, também são equivalentes, e assim por diante. Dados que os dois modelos foram construídos de forma idêntica, exceto pela função objetivo, os resultados dos testes de consistência feitos de 1.1 a 3.2 podem ser estendidos para os demais experimentos 4.1 a 6.2.

No primeiro de cada par de experimentos (x.1), a quantidade de mão de obra disponível ($a_l = 5$) excede a necessidade, permitindo ao *solver* encontrar soluções com as manutenções das aeronaves feitas em paralelo, ou seja, de forma que a quantidade desse recurso não restringisse as soluções. No segundo experimento do par (x.2), a quantidade de mão de obra disponível ($a_l = 1$) foi restringida de forma que todas as atividades precisem ser feitas em série, ou seja, não sendo possível a alocação de tarefas em paralelo. Dessa forma, foi possível avaliar o respeito aos limites de quantidades de mão de obra e às restrições de precedências. Outro ponto importante que a restrição da mão de obra permitiu avaliar foi a dificuldade do *solver* para encontrar soluções para o modelo, já que com uma quantidade menor deste recurso existem menos soluções viáveis. O experimento 3.2b foi um experimento adicional à parte, não possuindo equivalente, e será melhor detalhado mais ao fim desta seção.

As instâncias desta fase de experimentos possuem, 18 tarefas ($j = 1, \dots, 18$). Para todos os experimentos de teste, por simplicidade, todas as tarefas tiveram duração de duas unidades de tempo ($p_j = 2$). A mão de obra disponível foi de apenas um tipo ($w = 1$), também por simplicidade. No primeiro experimento de cada par (x.1), a MOD foi igual a cinco pessoas ($a_l = 5$) e no segundo (x.2), igual a uma pessoa ($a_l = 1$). Cada tarefa a ser executada demanda 1 pessoa ($n_{jw} = 1$), ou seja, $n_{jl} = 1$. Em todos os experimentos $r_j = 1$, ou seja, todas as tarefas estão imediatamente disponíveis para execução, restritas apenas pela precedência. As Tabelas 4.2 e 4.3 resumem os dados dos resultados de todos os experimentos de teste realizados. A Figura 4.1 ilustra a rede de precedência das instâncias com uma aeronave ($V = 1$) com duas tarefas fictícias adicionadas. Trata-se da mesma rede da Figura 3.1, no entanto foram adicionadas duas tarefas fictícias, a primeira ($j = 1$) e última tarefa da rede ($j = 18$), essas tarefas tem duração igual a zero ($p_1 = 0$ e $p_{18} = 0$).

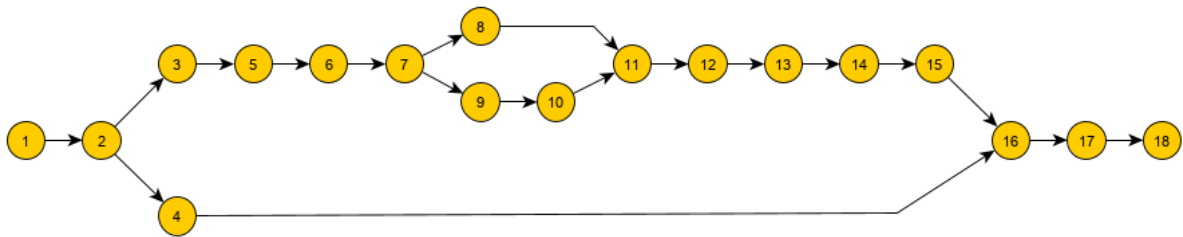


Figura 4.1: Rede de precedências para os problemas com uma aeronave.

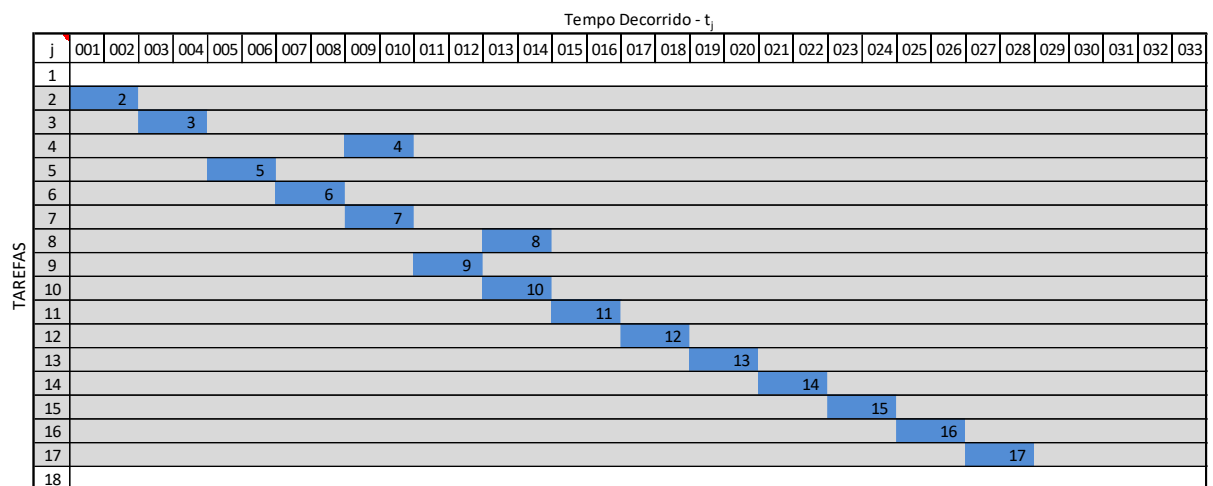


Figura 4.2: Gráfico de Gantt da solução encontrada para a instância com uma aeronave, teste 1.1 (minimizar o *leadtime* médio, $a_l = 5$).

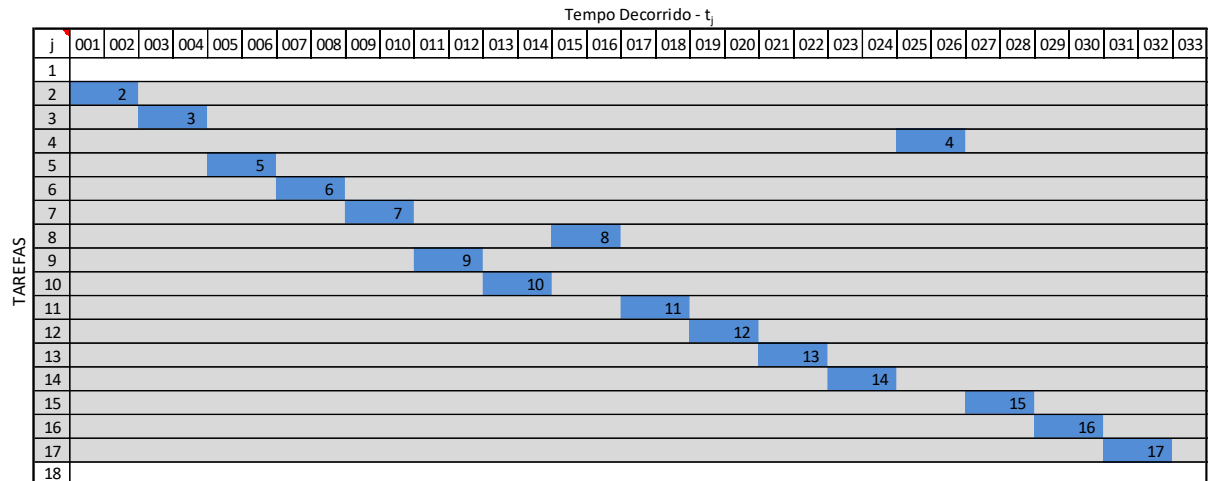


Figura 4.3: Gráfico de Gantt da solução encontrada para a instância com uma aeronave, teste 1.2 (minimizar o *leadtime* médio, $a_1 = 1$).

As Figuras 4.2 e 4.3 mostram como gráficos de Gantt as soluções encontradas pelo *solver* para o par de experimentos de apenas uma aeronave (testes 1.1 e 1.2). Nestas instâncias de experimentos foi definido T igual a 35 períodos ($T = 35$), número superior ao que seria o caminho crítico. Cada linha das Figuras 4.2 e 4.3 representa uma tarefa j do problema (primeira coluna das figuras) e, horizontalmente cada quadrado da primeira linha das figuras representa uma unidade de tempo (eixo do tempo decorrido). As unidades de tempo são tratadas apenas como tal, pois como são problemas totalmente fictícios, não faz diferença se são minutos, horas ou dias. Os blocos escuros representam a execução da tarefa, o momento em que cada tarefa do problema foi iniciada e terminada e, com isso, sua duração. Dentro de cada bloco o número marcado identifica o número da tarefa dentro da rede de precedência, Figura 4.1.

Na leitura dos períodos deve-se sempre considerar aqueles à direita do traço vertical no eixo do tempo decorrido. Sendo assim, a atividade 2 ($j = 2$) da Figura 4.2, por exemplo, inicia no período 1 ($t = 1$) e termina no período 3 ($t = 3$). O término da tarefa 17 ($j = 17$) da Figura 4.2, se projetado ao eixo do tempo decorrido, indicará o traço entre os períodos 28 e 29. A leitura correta é de que essa atividade teve sua execução concluída no instante de tempo 29.

Fica claro pela Figura 4.2, solução do teste.1.1, que com mão de obra abundante o *solver* usou todo o paralelismo possível na execução das atividades da rede, minimizando a função objetivo. Tanto o *leadtime* quanto o *makespan* da solução foram 29, ou seja, a última tarefa é concluída no instante de tempo 29. Já na solução do teste 1.2, Figura 4.3, com mão de obra restrita ao máximo, o *solver* colocou todas as atividades executadas em série, uma na

sequência da outra, respeitando a mão de obra disponível. Tanto o *leadtime* quanto o *makespan* da solução foram iguais a 33. Note que nos experimentos 1.1 e 1.2 o *leadtime* e o *makespan* coincidem, isso acontece por que esses experimentos foram feitos com apenas uma aeronave ($V = 1$). No resultado do teste 1.1 pode-se notar que as tarefas 4 e 7, 8 e 10 foram executadas em paralelo. Na solução do problema 1.2, essas mesmas tarefas não estão mais em paralelo, levando o *makespan* e o *leadtime* médio da solução para 33, maior que o da solução anterior. Os tempos computacionais e os *gaps* de otimalidade estão apresentados na Tabela 4.2 adiante. Essas soluções são comprovadamente ótimas pelo CPLEX.

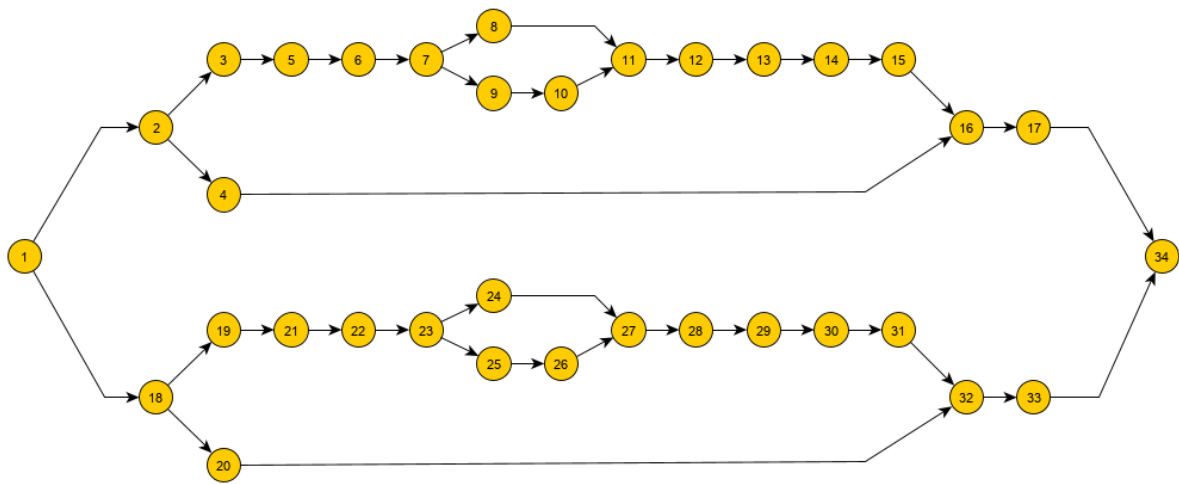


Figura 4.4: Rede de precedências para os problemas com duas aeronaves.

A Figura 4.4 acima ilustra a rede de precedências para os problemas com duas aeronaves a serem programadas. Para simplificar a programação, optou-se por numerar as atividades da rede de forma contínua. Dessa forma, a aeronave 1 é representada pelas atividades de 2 a 17, conforme a Figura 4.1, e a segunda aeronave é representada na rede pelas atividades de 18 a 33. As atividades 1 e 34 são as atividades fictícias de duração zero nessa rede ($p_1 = 0$ e $p_{34} = 0$), e marcam o início de todo o projeto e o seu término respectivamente. Vale notar que a primeira tarefa da aeronave 1 é a tarefa 2 ($j = 2$) e sua equivalente na aeronave 2 é a tarefa 18 ($j = 18$). Similarmente a equivalente da tarefa 11 ($j = 11$) na aeronave 1 é a tarefa 27 ($j = 27$) na aeronave 2, e assim para as demais tarefas.

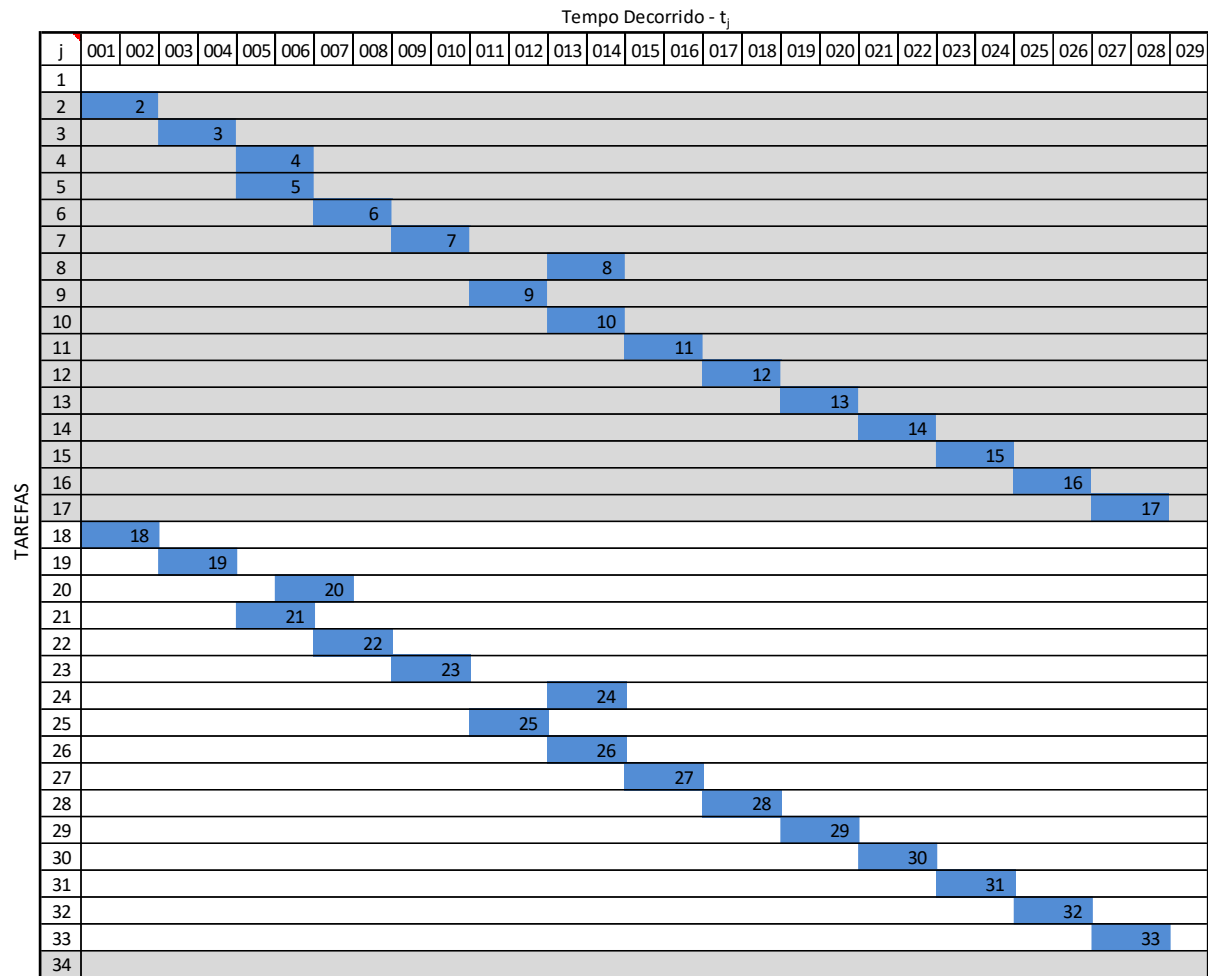


Figura 4.5: Gráfico de Gantt da solução encontrada para a instância com duas aeronaves, teste 2.1 (minimizar o *leadtime* médio, $a_1 = 5$).

A Figura 4.5 mostra o caso de duas aeronaves em paralelo, as regiões sombreadas e claras na figura diferenciam as duas aeronaves. Novamente o *solver* foi bem-sucedido para encontrar a solução ótima do problema com as duas aeronaves nessa situação. O *leadtime* médio e o *makespan* do Teste 2.1 foi de 29 unidades de tempo, conforme esperado, pois embora sejam 2 aeronaves, existe MOD suficiente para que ambas sejam executadas em paralelo. A Figura 4.6 a seguir mostra a solução do problema com duas aeronaves e mão de obra restrita e igual a uma pessoa. Nesse caso, as duas aeronaves foram colocadas em série, elevando o *makespan* para 65, o dobro da solução do Teste 1.2, também conforme esperado. O *leadtime* médio para este caso não foi igual ao *makespan* tendo sido igual a 49 unidades de tempo. A solução encontrada para este caso também foi ótima.

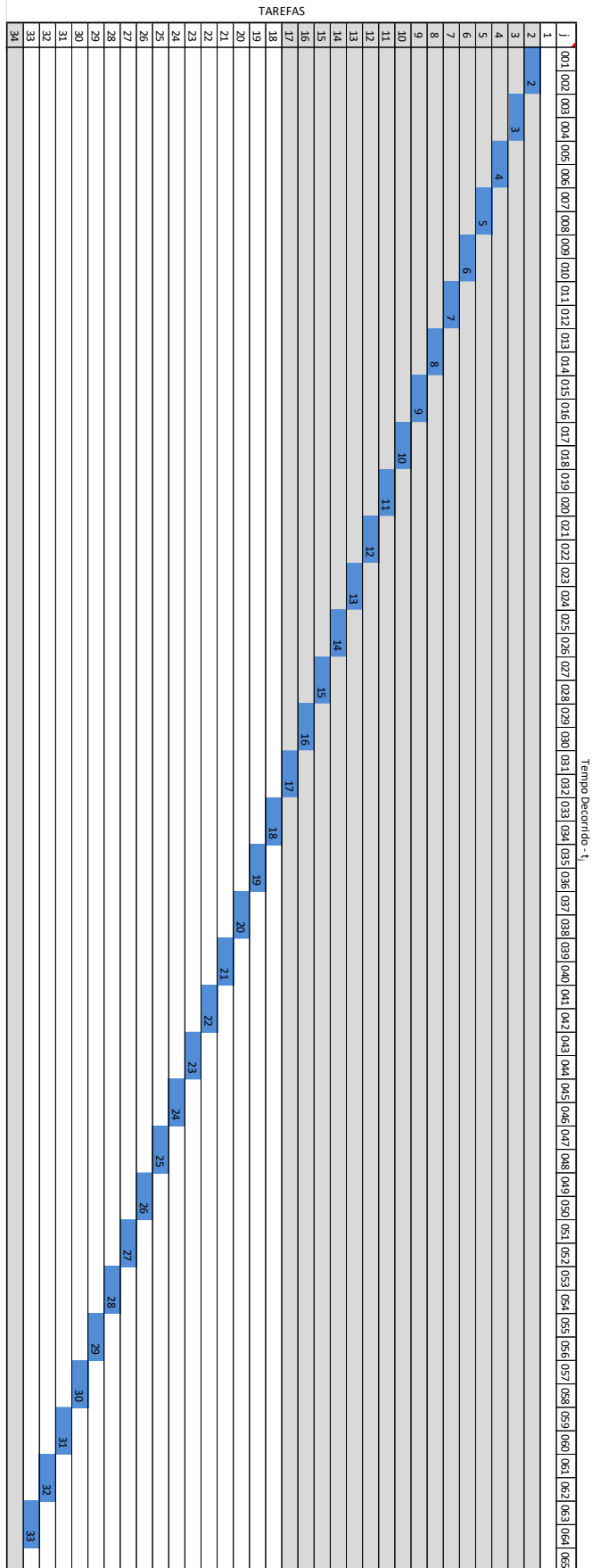


Figura 4.6: Gráfico de Gantt da solução para a instância com duas aeronaves em série, teste 2.2 (minimizar o *leadtime* médio, $a_1 = 1$).

A Figura 4.7 mostra a rede de precedências para o problema com quatro aeronaves. Ela segue as mesmas premissas já descritas para a rede com duas aeronaves. Vale observar que as primeiras tarefas de cada aeronave, de 1 a 4, são na rede respectivamente as tarefas $j = 2, j = 18, j = 34$ e $j = 50$. Nessa rede, as tarefas fictícias são as tarefas 1 e 66. Essas tarefas são, respectivamente, as referências de início e término de todo o projeto. Essa rede de quatro aeronaves é a rede que será utilizada também na próxima seção deste trabalho, que se debruçará sobre um problema com dados reais. Não há diferença nas redes da seção de testes e da seção de problemas reais.

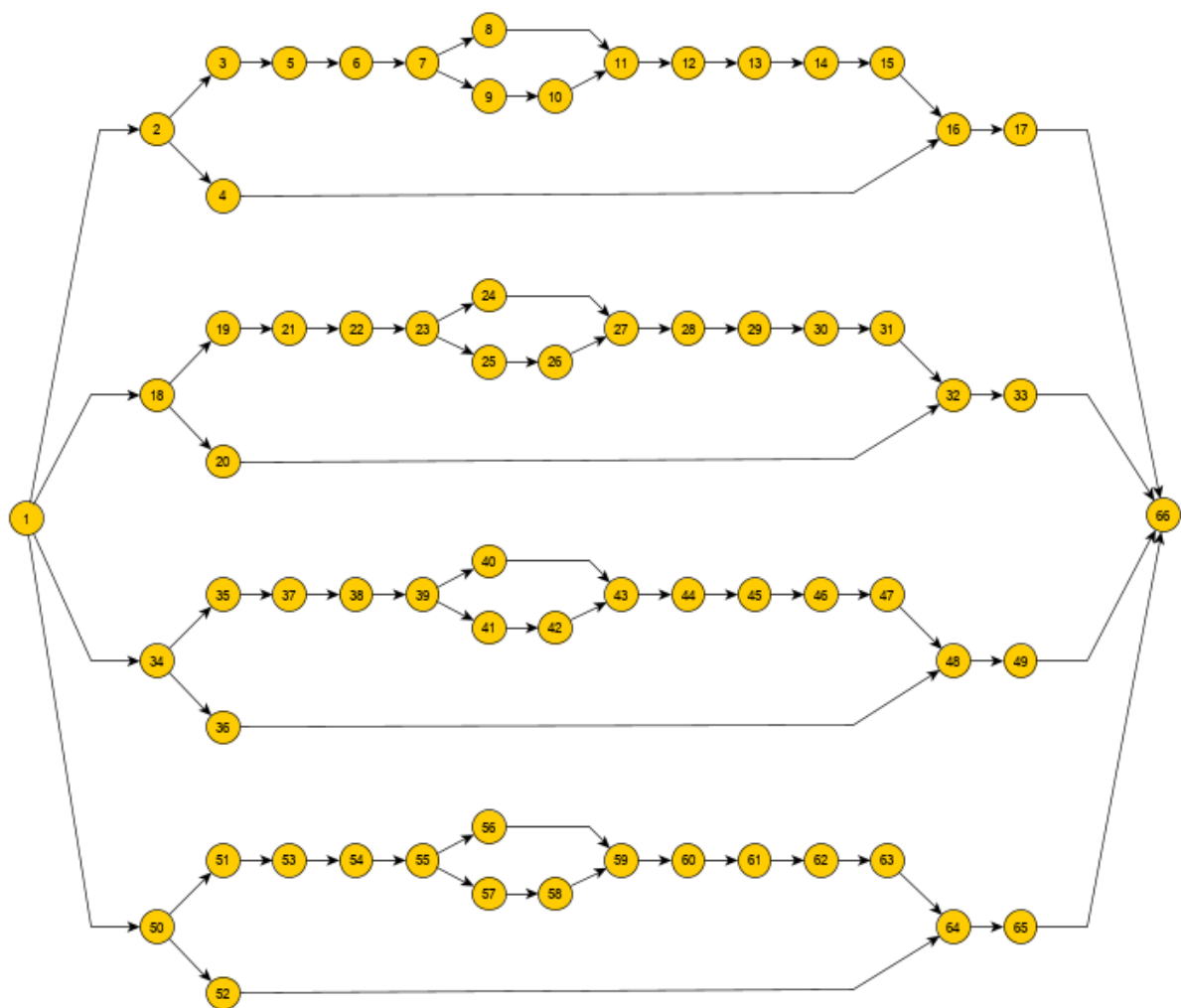


Figura 4.7: Rede de precedências para os problemas com quatro aeronaves.

A Figura 4.8 mostra o resultado encontrado pelo *solver* para o problema com quatro aeronaves. Novamente para o problema com recurso mão de obra abundante, o *solver* se comportou como nos demais testes, encontrando a solução ótima. Vale observar que nesse caso

o *leadtime* médio foi de 29,5, e não 29 como no caso do teste 2.1, similarmente o *makespan* da solução foi 31, e não 29 como no teste 2.1. Isso ocorreu porque em alguns momentos a solução atingiu o limite de MOD disponível de 5 pessoas (i.e. $a_l = 5$). Isso pode ser observado nos instantes 3 e 4 ($t = 3$ e $t = 4$), ou 13 e 14 ($t = 13$ e $t = 14$), por exemplo, em que existem 5 atividades sendo executadas ao mesmo tempo, cada uma utilizando uma pessoa.

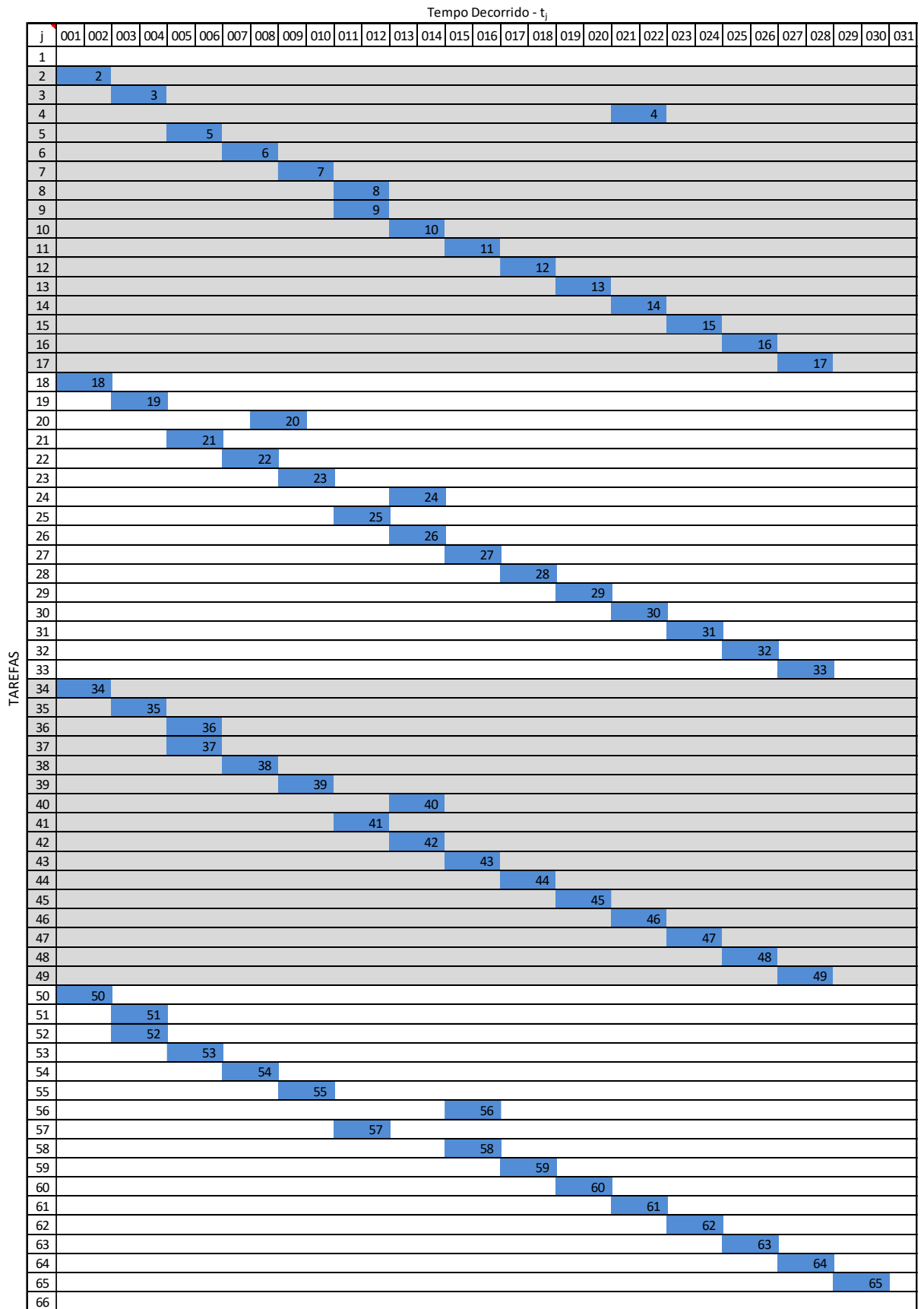


Figura 4.8: Gráfico de Gantt da solução para a instância com quatro aeronaves em paralelo, teste 3.1 (minimizar o *leadtime* médio, $a_1 = 5$).

O gráfico da Figura 4.9 mostra a quantidade de mão de obra ocupada por período no experimento de teste 3.1, com $a_1 = 5$. Conforme se pode notar em nenhum momento a solução do *solver* ultrapassou a quantidade de MOD disponível de 5 pessoas, evidenciando a formulação correta da restrição que limita a mão de obra. O gráfico da Figura 4.10 mostra a ocupação da mão de obra para a solução do teste 3.2b, com $a_1 = 1$, e novamente o modelo respeitou o limite de MOD disponibilizada. A solução do teste 3.2 é discutida em seguida.

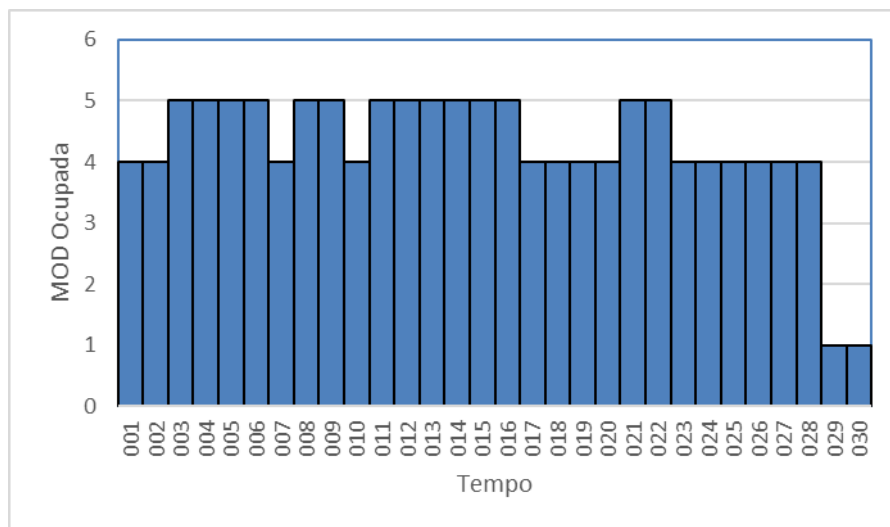


Figura 4.9: Ocupação da mão de obra no teste 3.1.

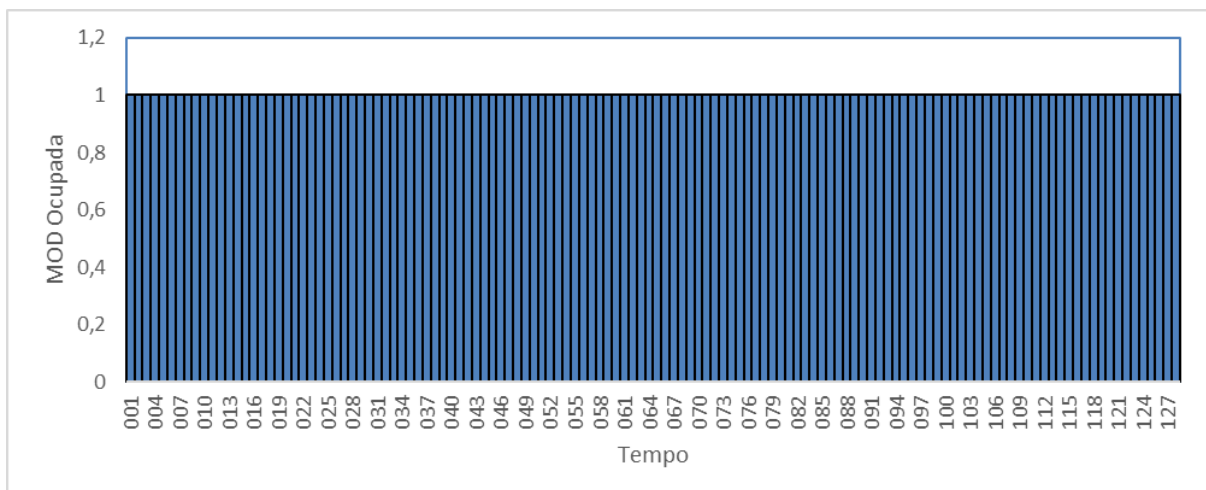


Figura 4.10: Ocupação da mão de obra no teste 3.2b.

Para o problema com mão de obra restrita e quatro aeronaves, teste 3.2, o *solver* não encontrou solução ótima dentro do limite de tempo estabelecido, que foi de 10 minutos

(600 segundos), e para $T = 130$. Isso não significa que não existe solução para o problema em questão, pois o experimento foi montado com a finalidade de testar o modelo, existindo então uma solução esperada. Foi ainda executado um teste aumentando o valor de T para 200 unidades de tempo ($T = 200$). A finalidade foi verificar se aumentando a gama de soluções possíveis para o problema, o *solver* chegaria a uma solução, que ainda que de baixa qualidade, fosse melhorada com o tempo de processamento. Isso elevou o número de variáveis e restrições para respectivamente, 12.076 e 339, e novamente não foi encontrada solução dentro do tempo limite.

A solução ótima só foi encontrada quando, alternativamente, foi fornecido um prazo de término (d_j) para as duas primeiras aeronaves do problema, teste 3.2b. Para esse problema, a solução esperada teria um *leadtime* médio de 81 e um *makespan* de 129. Como só foi fornecido recurso MOD para executar uma tarefa de cada vez, as quatro aeronaves deveriam ser concluídas nesse tempo, pois uma única aeronave é concluída em 32 unidades de tempo. É importante lembrar que, neste grupo de experimentos iniciais, o objetivo principal não foi verificar a capacidade do *solver* em encontrar soluções para os maiores problemas, mas sim de verificar se as soluções encontradas respeitam corretamente as restrições, e se o modelo foi formulado corretamente.

Dado que o problema é simples e a solução esperada é conhecida, para as duas primeiras aeronaves desse problema foi definido $d_{17} = 65$ e $d_{33} = 65$, ou seja, essas duas aeronaves se feitas em sequência, deveriam terminar as atividades no máximo na metade do tempo esperado para conclusão de todo o projeto. Para o teste 3.2b, o tempo de término do projeto foi definido como $T = 130$. A Figura 4.11 ilustra visualmente uma das soluções esperadas para o teste 3.2b, e por conseguinte, a lógica que embasou esse teste. Esse valor também foi estabelecido com base nos parâmetros conhecidos do problema, sendo que se todas as atividades das quatro aeronaves fossem feitas em sequência, 130 unidades de tempo seriam suficientes para conclusão de todo o projeto. Repare, na Figura 4.11, que embora a primeira aeronave precise de 32 unidades de tempo para ser executada, seu instante de término é o instante 33, a segunda o instante 65 e assim por diante. Isso acontece pois nesse tipo de modelo, o instante inicial é o período um, e não o período zero.

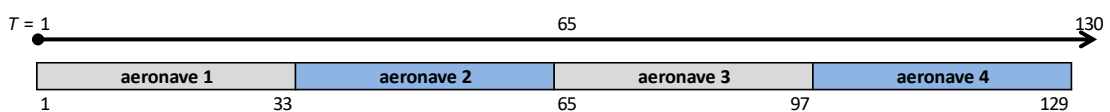


Figura 4.11: Solução esperada para o experimento teste 3.2.

Conforme mencionado, ao incluir uma restrição redundante adicional no problema ($d_{17} = 65$ e $d_{33} = 65$), sem perda de generalidade, usando os prazos d_j , o *solver* CPLEX foi capaz de encontrar uma solução ótima com o valor do *leadtime* médio 81, *gap* de otimalidade zero e *makespan* 129, em aproximadamente 23 segundos. A Figura 4.12 ilustra a solução encontrada. Outro ponto importante é que esse experimento permitiu avaliar o funcionamento do modelo quando fornecido um prazo d_j . Situação essa que pode ocorrer, por exemplo, caso na programação das aeronaves, se queira priorizar uma determinada aeronave. Os resultados mostram que isso é possível e o *solver* respeitou corretamente essa restrição colocada em d_j .

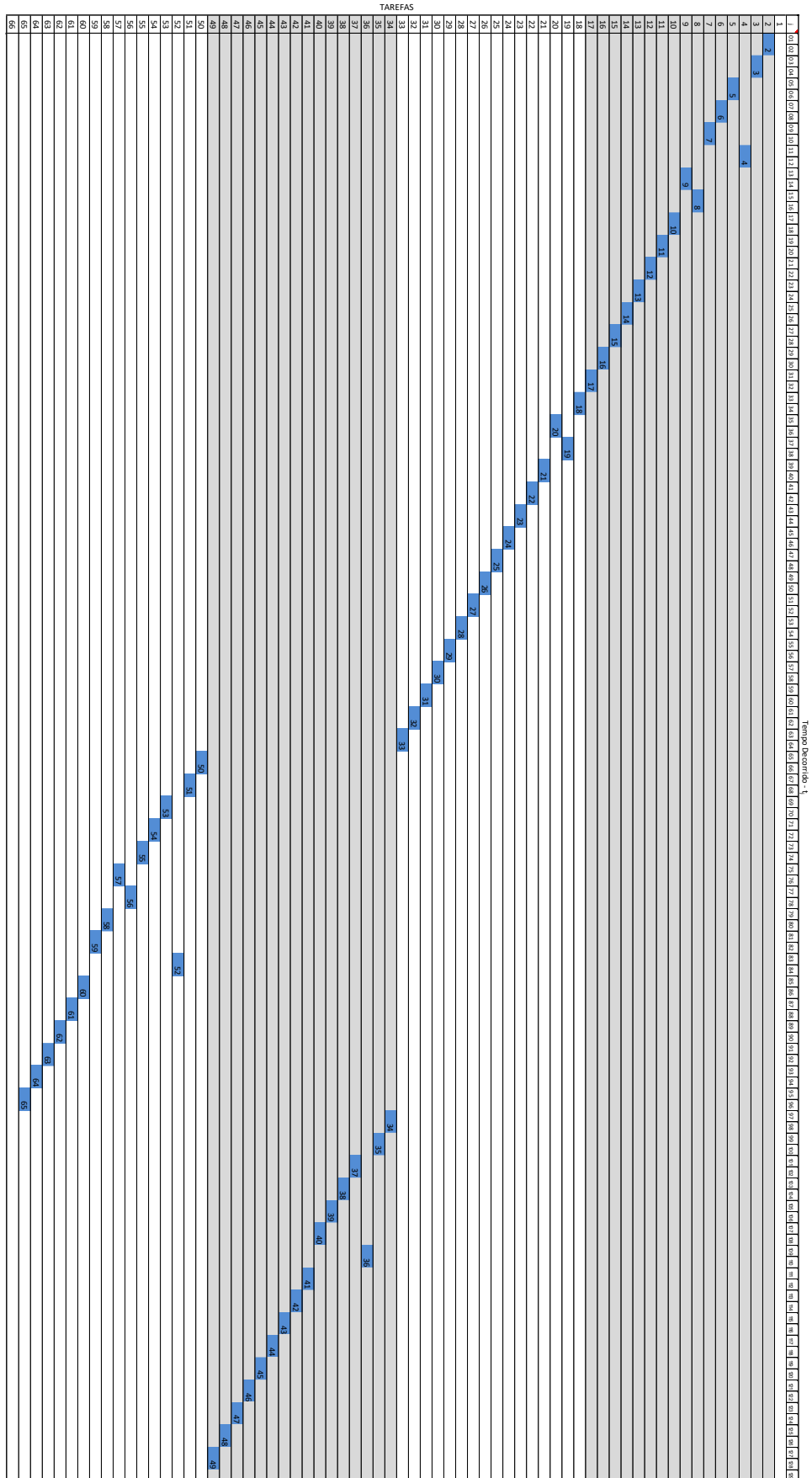


Figura 4.12: Gráfico de Gantt da solução para a instância com quatro aeronaves em série, teste 3.2b (minimizar o *leadtime* médio, $a_1 = 1$).

As Tabelas 4.2 e 4.3 mostram os resultados dos experimentos de testes para os modelos (13a)-(18) e (13b)-(18), respectivamente. Ambos os modelos (com funções objetivo diferentes) resultaram nos mesmos valores ótimos de *makespan*, para todos os testes. Note que podem resultar em valores diferentes em outros exemplos, dado que o primeiro modelo minimiza o *leadtime* médio e o segundo minimiza o *makespan*. Para o *leadtime* médio, os resultados são iguais em todas as instâncias exceto nos testes 3.1 e 6.1, em que o valores do *leadtime* médio são respectivamente 29,5 e 30,25. Embora no teste 6.1 o modelo tenha atingido a solução ótima para o *makespan*, note que o *leadtime* médio obtido não é o ótimo (i.e. 29,5). Isso mostra que o modelo que otimiza o *leadtime* médio pode levar a *makespans* ótimos, e o modelo que otimiza o *makespan* pode igualmente levar a *leadtime* médios ótimos, mas isso não é uma regra.

Tabela 4.2: Resultados dos experimentos de teste com função objetivo minimizar o *leadtime* médio.

Teste	<i>Leadtime</i> Médio (períodos)	<i>Makespan</i> (períodos)	Número de restrições	Número de variáveis	<i>Gap</i> (%)	Tempo (segundos)
1.1	29*	29	61	327	0	0,02
1.2	33*	33	67	329	0	0,03
2.1	29*	29	131	1.683	0	0,11
2.2	49*	65	134	1.683	0	1,17
3.1	29,5*	31	268	7.526	0	3,55
3.2	81	129	269	7.526	3,08	20,4
3.2b	81*	129	269	5.446	0	22,5

* solução comprovadamente ótima

Tabela 4.3: Resultados dos experimentos de teste com função objetivo minimizar o *makespan*.

Teste	<i>Leadtime</i> Médio (períodos)	<i>Makespan</i> (períodos)	Número de restrições	Número de variáveis	<i>Gap</i> (%)	Tempo (segundos)
4.1	29	29*	62	362	0	0,03
4.2	33	33*	68	364	0	0,08
5.1	29	29*	131	1.683	0	0,09
5.2	49	65*	134	1.683	0	1,26
6.1	30,25	31*	268	7.526	0	4,78
6.2	81	129*	269	7.526	0	19,19

* solução comprovadamente ótima

A Tabela 4.2 mostra que à medida que o tamanho das instâncias do problema aumentam, ocorrem aumentos significativos nas quantidades de variáveis e restrições, e no tempo computacional. Ao aumentar de uma para quatro aeronaves, o número de variáveis salta de pouco mais de trezentas para mais de sete mil e quinhentas variáveis, um aumento de mais de 2100%. Cabe notar a significativa queda no número de variáveis do teste 3.2b (para 5.446 variáveis), quando no experimento foi fornecido um d_j inicial para as duas primeiras aeronaves. Isso levou o *solver* a conseguir encontrar a solução ótima do problema em 22,5 segundos. Para as duas instâncias do teste 3.2, o *solver* encontrou soluções iguais, de *leadtime* médio igual a 81. Dessa forma, ao se verificar que na segunda instância do teste 3.2 o *leadtime* médio 81 possui *gap* de otimalidade zero, compreende-se que, na verdade, na primeira instância do teste 3.2 o *solver* chegou a uma solução ótima, no entanto, não conseguiu fechar o *gap* de otimalidade para zero dentro do tempo limite de execução.

O *gap* de otimalidade é uma medida da qualidade da solução encontrada pelo *solver*. A equação (20) abaixo mostra como o *solver* CPLEX 12 efetua o cálculo do *gap*. Seu valor é distância entre a melhor solução possível (*BP*) para o problema não discretizado e a melhor solução factível encontrada (*BF*) pelo *solver*, dividida pela própria solução (*BF*). A equação (20) e outros detalhes, podem ser encontrados na documentação de utilização e parâmetros do CPLEX 12 (CPLEX 12, 2019).

$$\frac{(|BP - BF|)}{(1e^{-10} + |BF|)} \quad (20)$$

A Tabela 4.3 mostra os resultados do segundo modelo (13b)-(18) para as instâncias equivalentes, no entanto com a função objetivo de minimizar o makespan diretamente. Neste grupo de experimentos o *solver* encontrou, para todos os casos, uma solução com *gap* de otimalidade zero. O crescimento do problema é um obstáculo maior para o *solver* quando a função objetivo é a de minimizar o *leadtime* médio. Para uma melhor compreensão deste ponto seriam necessários mais testes. Nesse trabalho, no entanto, não se aprofundará mais nessa questão, que ficará como perspectiva futura.

Outro ponto importante a ser notado é que, com apenas quatro aeronaves ao mesmo tempo, o *solver* CPLEX já apresentou alguma dificuldade para encontrar uma solução factível. No entanto, quando restringido o universo de soluções possíveis, por meio do fornecimento de um d_j inicial, para o modelo (13a)-(18), ele encontrou a solução ótima. O experimento 3.2 adicional com o fornecimento do d_j inicial foi feito apenas para esse modelo,

no caso do modelo (13b)-(18), o solver encontrou a solução ótima. Esse pode ser um caminho a seguir com os problemas reais, já que esses tendem a ser maiores e mais complexos, e uma solução inicial como ponto de partida para o *solver* pode ser obtida, assim como na prática.

Com os experimentos testes realizados, é possível avaliar satisfatoriamente as formulações (13a)-(18) e (13b)-(18) do modelo nesta etapa. O *solver* CPLEX foi capaz de gerar soluções factíveis, sendo que para os problemas mais simples ele o fez em frações de segundos. As soluções geradas respeitaram corretamente, em todos os casos, as restrições de precedência e disponibilidade de mão de obra. Isso pode ser visto com clareza na Figura 4.8, solução do teste 3.1. A formulação parece adequada ao problema e capaz de gerar soluções satisfatórias, e as funções objetivo atendem ao proposto. Conforme esperado, o *solver* CPLEX teve dificuldades de encontrar soluções para problemas com restrições mais severas dos recursos. Para esta fase e seus objetivos, o modelo atingiu o objetivo proposto. A questão relacionada ao tamanho do problema é melhor avaliada nos próximos experimentos.

4.2 EXPERIMENTOS COM UM PROBLEMA REAL

Nessa seção foram estudadas instâncias do problema com dados reais. O principal objetivo dessa etapa foi explorar um problema com dados e dimensões reais, e avaliar o comportamento do *solver* ao resolver os modelos na busca por soluções. Foi simulada a chegada de quatro aeronaves no centro de manutenção ($V = 4$ ou $v = 1, \dots, 4$), mesma precedência dos experimentos anteriores, mas com intervalos de uma hora de diferença, diferentemente dos testes da seção anterior. Essas aeronaves realizariam todas o mesmo *check* de manutenção, o mais simples do plano de manutenções, equivalente a uma parada de 6 MO. Foram consideradas todas as tarefas planejadas constantes no pacote de manutenção. As atividades não planejadas, sem um pacote de horas de trabalho definidas, foram descartadas. No entanto, as atividades ainda que consideradas não planejadas previamente, por não fazerem parte do pacote básico de manutenção, mas com uma quantidade de horas de trabalho definidas, foram sim consideradas, pois na visão deste trabalho elas são planejáveis.

Foi utilizado para isso um pacote de manutenção real da empresa, e os dados foram coletados do sistema computacional da empresa. Foi aplicado aos dados um fator multiplicador, de forma a proteger os interesses da empresa. Vindo de problemas reais, a quantidade de períodos do problema será maior, e próxima destes. Os dados foram extraídos

em quantidade de horas homem necessárias para a execução das atividades, e convertidos em períodos de tempo corridos.

O pacote de trabalho foi estimado para ser executado conforme o processo de planejamento da empresa, com base na quantidade de horas homem. É possível extrair do sistema da empresa a quantidade de hora homem, de cada tipo de mão de obra, necessária em cada etapa da rede da Figura 3.1 ou 4.1 apresentadas anteriormente. A partir de um objetivo, um prazo médio para a realização do *check* de manutenção, é possível calcular o esforço de mão de obra necessário por unidade de tempo para que se cumpra tal prazo.

Neste experimento, conforme o problema real, foram necessários três tipos de mão de obra diferentes $W = 3$, e para cada um dos nós da rede foram calculadas as quantidades necessárias de pessoas de cada tipo de especialidade. A quantidade de períodos utilizada foi de 400 horas ou períodos, $T = 400$, quantidade um pouco superior à quantidade de horas necessárias para realizar as atividades das quatro aeronaves em série seguindo o caminho crítico.

A Tabela 4.4 mostra os parâmetros de cada uma das tarefas do problema e as quantidades de mão de obra de cada tipo, requeridas para as atividades. A rede de atividades para esse problema (Figura 4.7), está representada pelas atividades 1 a 66 ($J = 66$). A tarefa 1 é uma tarefa fictícia de duração zero (i.e. $p_1 = 0$) e MOD zero (i.e. $n_{11} = 0$, $n_{12} = 0$, $n_{13} = 0$), que tem o objetivo apenas de servir de marco inicial para o projeto. A tarefa 66, similarmente, também com duração zero e MOD zero, serve como marco final do projeto. Vale notar que alguns nós da rede têm duração zero na Tabela 4.4, o que significa que não existem tarefas a serem executadas nessas fases para essa parada de manutenção. Outro ponto é a quantidade de mão de obra requerida para cada etapa j dos tipos 1 (n_{j1}), 2 (n_{j2}) e 3 (n_{j3}), que é diferente para cada etapa, repetindo-se apenas para as mesmas etapas de aeronaves diferentes.

Tabela 4.4: Dados de entrada do problema com dados reais.

j	p_j	n_{j1}	n_{j2}	n_{j3}	r_j
1	0	0	0	0	0
2	9	2	2	2	1
3	4	3	1	1	10
4	1	2	1	0	10
5	0	0	0	0	14
6	0	0	0	0	14
7	19	1	4	2	14
8	5	1	3	2	33

9	1	0	0	4	33
10	1	0	0	2	34
11	0	0	0	0	38
12	7	2	4	0	38
13	18	0	3	2	45
14	9	0	2	3	63
15	0	0	0	0	72
16	3	1	1	3	72
17	0	0	0	0	75
18	9	2	2	2	2
19	4	3	1	1	11
20	1	2	1	0	11
21	0	0	0	0	15
22	0	0	0	0	15
23	19	1	4	2	15
24	5	1	3	2	34
25	1	0	0	4	34
26	1	0	0	2	35
27	0	0	0	0	39
28	7	2	4	0	39
29	18	0	3	2	46
30	9	0	2	3	64
31	0	0	0	0	73
32	3	1	1	3	73
33	0	0	0	0	76
34	9	2	2	2	3
35	4	3	1	1	12
36	1	2	1	0	12
37	0	0	0	0	16
38	0	0	0	0	16
39	19	1	4	2	16
40	5	1	3	2	35
41	1	0	0	4	35
42	1	0	0	2	36
43	0	0	0	0	40
44	7	2	4	0	40
45	18	0	3	2	47
46	9	0	2	3	65
47	0	0	0	0	74
48	3	1	1	3	74
49	0	0	0	0	77
50	9	2	2	2	4
51	4	3	1	1	13
52	1	2	1	0	13
53	0	0	0	0	17

54	0	0	0	0	17
55	19	1	4	2	17
56	5	1	3	2	36
57	1	0	0	4	36
58	1	0	0	2	37
59	0	0	0	0	41
60	7	2	4	0	41
61	18	0	3	2	48
62	9	0	2	3	66
63	0	0	0	0	75
64	3	1	1	3	75
65	0	0	0	0	78
66	0	0	0	0	0

Na formulação destes experimentos foi fornecido um r_j , inicial para cada tarefa, que pode ser visto na coluna r_j da Tabela 4.4. Uma vez que as 4 aeronaves chegam com intervalos de 1 hora e seguem uma rede precedência definida, foi possível calcular, por meio do CPM (*Critical Path Method*), o instante mais cedo em que cada tarefa j pode ser iniciada. Esse é um passo importante pois permite ao *solver*, descartar todas as possíveis soluções com r_j inferior a tais valores, melhorando o desempenho ao reduzir o conjunto das possíveis soluções a serem analisadas. Isso é feito por meio da restrição (17) dos modelos.

O estudo consistiu em resolver o modelo com mão de obra abundante, e uma vez encontrada a solução ótima, foram executadas novas instâncias reduzindo gradativamente a quantidade de mão de obra disponível, o que aumentou a dificuldade do *solver* de encontrar uma solução ótima ou factível. Isso foi feito até que o *solver* não conseguisse mais encontrar uma solução ótima para o problema, aumentando significativamente o *gap* de otimalidade.

Outra análise feita foi executar o modelo com base na mão de obra calculada pelo planejamento da empresa. Isso permitiu, por meio de comparações com todos os resultados obtidos, analisar soluções alternativas e melhores com base na MOD disponível, e inferir outras possíveis soluções para o problema, que foram experimentadas e analisadas. Todos os experimentos dessa fase do trabalho foram executados com um tempo limite no *solver* de 3 horas ou 10.800 segundos, tempo esse considerado razoável para obtenção de uma solução na rotina da empresa quando a programação precisa ser feita ou revisada. Uma vez atingido esse tempo, o *solver* foi interrompido e a melhor solução factível foi analisada. Os problemas nessa fase são maiores que na fase anterior. Os modelos possuem 3.947 restrições e 26.401 variáveis. Após a execução do *presolver*, o CPLEX reduz as restrições em, aproximadamente, 69% e as variáveis em 19%, para em média 1.251 e 21.430, respectivamente.

As Tabelas 4.5 e 4.6 mostram os resultados desses experimentos. Foram realizados ao todo trinta e sete experimentos com o mesmo problema para cada função objetivo, o que totalizou setenta e quatro experimentos, com a finalidade de explorar a complexidade desses problemas, e a capacidade do *solver* de gerar soluções factíveis para os modelos em tempos computacionais aceitáveis. A Tabela 4.5 apresenta os resultados obtidos com o modelo (13a)-(18); esses experimentos foram identificados como 7.1 a 7.37, e a Tabela 4.6 mostra os resultados obtidos com o modelo (13b)-(18), que foram identificados de 8.1 a 8.37. Os problemas 7.1 e 8.1 são equivalentes, assim como o 7.2 e 8.2, e assim por diante, diferenciando-se apenas pela função objetivo do modelo. Dentre os experimentos x.1 e x.37, o que mudou foram as quantidades de MOD disponíveis para que o *solver* fizesse a programação das tarefas. Essas quantidades são mostradas nas colunas a_1 , a_2 , e a_3 das Tabelas 4.5 e 4.6.

Tabela 4.5: Resultados dos experimentos função objetivo min *leadtime* médio, modelo (13a)-(18).

Exp.	Leadtime Médio (períodos)	Makespan		Gap (%)	a_1	a_2	a_3	MOD total	Ocup. Média a_1	Ocup. Média a_2	Ocup. Média a_3	Ocup. Média Total	Tempo (segundos)
		(períodos)	(dias)										
7.1	76,5	78	5,6	0	18	18	18	54	21%	63%	43%	42%	1,50
7.2	76,5	78	5,6	0	17	17	17	51	22%	66%	45%	45%	1,94
7.3	76,5	78	5,6	0	16	16	16	48	24%	70%	48%	47%	1,89
7.4	81,8	94	6,7	0	15	15	15	45	21%	62%	42%	42%	32,56
7.5	82,8	94	6,7	0	14	14	14	42	22%	67%	45%	45%	61,30
7.6	84,0	106	7,6	0	13	13	13	39	21%	64%	43%	43%	187,81
7.7	86,8	100	7,1	0	12	12	12	36	25%	73%	50%	49%	2.790
7.8	95,5	131	9,4	3,7	11	11	11	33	20%	61%	41%	41%	2.145
7.9	98,0	127	9,1	4,5	10	10	10	30	23%	69%	47%	46%	7.010
7.10	104,5	142	10,1	8,2	9	9	9	27	23%	68%	47%	46%	10.255
7.11	103,3	137	9,8	0	8	8	8	24	27%	80%	54%	54%	3.808
7.12	125,8	160	11,4	9,9	7	7	7	21	26%	78%	53%	52%	10.270
7.13	147,5	212	15,1	12,5	6	6	6	18	23%	69%	47%	46%	10.559
7.14	169,3	257	18,4	12,3	5	5	5	15	23%	70%	48%	47%	2.087
7.15	178,8	274	19,6	12,7	4	4	4	12	27%	79%	54%	53%	10.699
7.16	105,3	139	9,9	4,7	7	8	8	23	30%	79%	54%	55%	9.970
7.17	129,5	184	13,1	12,4	8	7	8	23	20%	69%	41%	42%	10.167
7.18	118,0	164	11,7	15,2	8	8	7	23	23%	67%	52%	47%	10.370
7.19	104,3	138	9,9	3,6	6	8	8	22	36%	79%	54%	58%	7.900
7.20	109,0	150	10,7	7,2	5	8	8	21	39%	73%	50%	56%	10.599
7.21	112,8	153	10,9	10,0	4	8	8	20	48%	71%	49%	58%	10.472
7.22	133,5	160	11,4	15,2	7	7	8	22	26%	78%	47%	50%	10.545
7.23	128,0	163	11,6	11,6	6	7	7	20	30%	77%	52%	54%	10.068
7.24	151,0	218	15,6	14,4	6	6	7	19	23%	67%	39%	43%	10.701
7.25	126,0	175	12,5	9,5	4	7	7	18	42%	71%	49%	56%	10.671

7.26	102,8	129	9,2	11,6	4	12	8	24	57%	57%	58%	57%	10.506
7.27	99,5	128	9,1	6,6	6	10	8	24	38%	68%	58%	57%	10.066
7.28	103,3	137	9,8	11,1	4	11	8	23	54%	58%	54%	56%	10.265
7.29	101,3	137	9,8	7,5	5	10	8	23	43%	64%	54%	56%	10.043
7.30	109,0	142	10,1	15,6	4	11	7	22	52%	56%	60%	56%	10.700
7.31	111,0	150	10,7	15,8	5	10	7	22	39%	58%	57%	53%	10.606
7.32	112,0	133	9,5	16,3	4	10	7	21	55%	66%	64%	63%	10.448
7.33	119,8	160	11,4	13,7	3	10	7	20	61%	55%	53%	55%	9.868
7.34	117,0	137	9,8	19,3	4	10	6	20	54%	58%	54%	56%	10.557
7.35	128,5	161	11,5	19,6	3	10	6	19	61%	54%	62%	58%	10.114
7.36	114,3	157	11,2	14,2	4	9	6	19	47%	62%	63%	59%	10.800
7.37	126,5	179	12,8	17,1	3	9	6	18	55%	54%	55%	55%	10.669

Tabela 4.6: Resultados dos experimentos função objetivo min *makespan*, modelo (13b)-(18).

Exp.	Leadtime Médio (períodos)	Makespan		Gap (%)	a_1	a_2	a_3	MOD total	Ocup. Média a_1	Ocup. Média a_2	Ocup. Média a_3	Ocup. Média Total	Tempo (segundos)
		(períodos)	(dias)										
8.1	76,5	78	5,6	0	18	18	18	54	21%	63%	43%	42%	2,69
8.2	76,5	78	5,6	0	17	17	17	51	22%	66%	45%	45%	2,56
8.3	76,5	78	5,6	0	16	16	16	48	24%	70%	48%	47%	2,48
8.4	86,3	94	6,7	0	15	15	15	45	21%	62%	42%	42%	24,06
8.5	90,5	94	6,7	0	14	14	14	42	22%	67%	45%	45%	112,83
8.6	93,3	94	6,7	0	13	13	13	39	24%	72%	49%	48%	84,00
8.7	90,5	94	6,7	0	12	12	12	36	26%	78%	53%	52%	100,80
8.8	102,0	106	7,6	0	11	11	11	33	25%	75%	51%	51%	139,56
8.9	108,0	109	7,8	0	10	10	10	30	27%	80%	55%	54%	2.936
8.10	119,3	120	8,6	17,7	9	9	9	27	27%	81%	55%	55%	1.596
8.11	116,0	126	9,0	0	8	8	8	24	29%	87%	59%	58%	10.119
8.12	141,5	153	10,9	22,2	7	7	7	21	28%	83%	56%	56%	2.818
8.13	158,0	176	12,6	26,2	6	6	6	18	28%	83%	56%	56%	10.264
8.14	217,5	232	16,6	35,7	5	5	5	15	26%	76%	52%	51%	10.771
8.15	255,0	276	19,7	42,1	4	4	4	12	27%	80%	54%	54%	10.687
8.16	125,3	129	9,2	18,3	7	8	8	23	33%	85%	58%	60%	1.794
8.17	140,5	151	10,8	22,1	8	7	8	23	24%	83%	49%	51%	9.930
8.18	118,0	128	9,1	17,1	8	8	7	23	29%	85%	67%	60%	5.272
8.19	131,0	133	9,5	21,0	6	8	8	22	37%	82%	56%	60%	9.367
8.20	132,0	136	9,7	22,2	5	8	8	21	44%	81%	55%	62%	7.368
8.21	143,3	146	10,4	26,3	4	8	8	20	50%	75%	51%	60%	6.246
8.22	145,0	152	10,9	20,0	7	7	8	22	28%	82%	49%	53%	10.478
8.23	141,3	151	10,8	22,9	6	7	7	20	32%	83%	56%	58%	4.162
8.24	160,8	177	12,6	26,7	6	6	7	19	28%	82%	48%	52%	8.034
8.25	139,3	155	11,1	25,0	4	7	7	18	47%	81%	55%	63%	10.295
8.26	113,3	116	8,3	19,0	4	12	8	24	64%	63%	65%	64%	10.304
8.27	109,3	113	8,1	13,3	6	10	8	24	43%	78%	66%	65%	10.782

8.28	107,5	117	8,4	20,1	4	11	8	23	63%	68%	64%	66%	10.451
8.29	114,3	118	8,4	20,7	5	10	8	23	50%	74%	63%	65%	10.063
8.30	116,8	125	8,9	21,3	4	11	7	22	59%	64%	69%	65%	5.000
8.31	108,8	120	8,6	19,9	5	10	7	22	49%	73%	71%	67%	2.424
8.32	126,3	131	9,4	25,5	4	10	7	21	56%	67%	65%	64%	2.384
8.33	141,3	148	10,6	25,7	3	10	7	20	67%	59%	58%	60%	10.490
8.34	136,8	139	9,9	27,8	4	10	6	20	54%	64%	73%	64%	10.416
8.35	133,5	144	10,3	23,5	3	10	6	19	68%	61%	69%	64%	10.731
8.36	133,0	138	9,9	26,1	4	9	6	19	54%	71%	73%	68%	10.729
8.37	141,5	153	10,9	29,9	3	9	6	18	64%	63%	65%	64%	10.083

Os experimentos nessa fase foram iniciados com quantidade abundante de mão de obra, no intuito de que fosse obtida uma solução ótima. É possível calcular a quantidade de mão de obra ideal para a rede da Figura 4.1, com os dados da Tabela 4.4. Com um tempo padrão para esse *check* de 75 horas úteis (aproximadamente 5 dias úteis), e considerando 14 horas úteis por pessoa por dia, é obtida a seguinte configuração ideal de técnicos (MOD) por especialidades por dia requeridos: $a_1 = 1$, $a_2 = 3$, $a_3 = 2$. Pelos cálculos da área de planejamento, essa configuração de distribuição da mão de obra especializada é suficiente para realizar a referida manutenção de uma aeronave em aproximadamente 5 dias. Logo, se a empresa desejar liberar as 4 aeronaves no mínimo prazo deverá alocar 4 vezes essa mão de obra para atingir tal objetivo.

Se forem totalizadas as quantidades de MOD do raciocínio acima exposto se chegará numa demanda de 24 técnicos no total. De posse desse número, optou-se nesse trabalho, por iniciar os testes com uma quantidade de técnicos um pouco acima do dobro da quantidade necessária, 54 técnicos. Esses valores foram divididos igualmente entre as três especialidades, resultando em 18 pessoas por especialidade no experimento 7.1 e 8.1. Os experimentos foram então repetidos, reduzindo-se a mão de obra disponível em 1 pessoa por especialidade, até o limite de 4 pessoas por especialidade. Isso foi feito com a intenção de obter a melhor solução possível do *solver* e na sequência, sobrecarregando o mesmo, para que se pudesse verificar até que ponto este seria capaz de gerar soluções ótimas, e analisar a qualidade das soluções em sobrecarga.

Nos dois modelos até a configuração de 12 pessoas por especialidade, experimentos 7.1 a 7.7 e 8.1 a 8.7, as soluções de *gap* zero foram encontradas. No modelo da função objetivo de minimizar o *makespan*, o *solver* ainda conseguiu chegar a uma solução ótima para as instâncias 8.8 e 8.9 com, respectivamente, 11 e 10 MOD de cada tipo. Após interromper um ciclo de soluções ótimas encontradas (7.8, 7.9, 7.10 e 8.10), ambos os modelos encontraram

as soluções ótimas para o problema com 8 MOD de cada tipo (7.11 e 8.11), experimentos com o número absoluto de 24 MOD, calculado como ideal. Esses resultados reforçam o que já foi observado na fase de testes anteriores de que o *solver* com o modelo *makespan* (13b)-(18) possui uma melhor capacidade, ainda que pequena, de encontrar soluções de *gap* zero, do que o modelo do *leadtime* médio (13a)-(18).

A Tabela 4.7 faz um comparativo entre os valores das funções objetivo de cada modelo e seus equivalentes no outro modelo (os valores são os mesmos das Tabelas 4.5 e 4.6). Essa tabela tem o objetivo apenas de facilitar a comparação. Pode-se notar que para os experimentos x.1 a x.5, para os quais obteve-se as soluções ótimas para os dois modelos, foram obtidos valores iguais de *makespan*. O modelo do *leadtime* médio, embora tenha chegado ao *leadtime* médio (função objetivo) ótimo nesses casos, chegou aos mesmos *makespans* ótimos obtidos pelo modelo do *makespan*. Vale notar que, nesses casos, a MOD disponível de cada atividade era abundante. O oposto também ocorre, mas o modelo *makespan* obteve o *leadtime* médio ótimo apenas nos experimentos 8.1 a 8.3.

Tabela 4.7: Comparativo entre os valores das funções objetivo de cada modelo e seus equivalentes.

Exper.	<i>Leadtime</i> Médio (períodos)		<i>Makespan</i> (períodos)		<i>Gap</i> de otimalidade (%)	
	(13a)-(18)	(13b)-(18)	(13a)-(18)	(13b)-(18)	(13a)-(18)	(13b)-(18)
x.1	76,5	76,5	78	78	0	0
x.2	76,5	76,5	78	78	0	0
x.3	76,5	76,5	78	78	0	0
x.4	81,8	86,3	94	94	0	0
x.5	82,8	90,5	94	94	0	0
x.6	84,0	93,3	106	94	0	0
x.7	86,8	90,5	100	94	0	0
x.8	95,5	102,0	131	106	3,7	0
x.9	98,0	108,0	127	109	4,5	0
x.10	104,5	119,3	142	120	8,2	17,7
x.11	103,3	116,0	137	126	0	0
x.12	125,8	141,5	160	153	9,9	22,2
x.13	147,5	158,0	212	176	12,5	26,2
x.14	169,3	217,5	257	232	12,3	35,7
x.15	178,8	255,0	274	276	12,7	42,1
x.16	105,3	125,3	139	129	4,7	18,3
x.17	129,5	140,5	184	151	12,4	22,1
x.18	118,0	118,0	164	128	15,2	17,1
x.19	104,3	131,0	138	133	3,6	21,0
x.20	109,0	132,0	150	136	7,2	22,2
x.21	112,8	143,3	153	146	10,0	26,3

x.22	133,5	145,0	160	152	15,2	20,0
x.23	128,0	141,3	163	151	11,6	22,9
x.24	151,0	160,8	218	177	14,4	26,7
x.25	126,0	139,3	175	155	9,5	25,0
x.26	102,8	113,3	129	116	11,6	19,0
x.27	99,5	109,3	128	113	6,6	13,3
x.28	103,3	107,5	137	117	11,1	20,1
x.29	101,3	114,3	137	118	7,5	20,7
x.30	109,0	116,8	142	125	15,6	21,3
x.31	111,0	108,8	150	120	15,8	19,9
x.32	112,0	126,3	133	131	16,3	25,5
x.33	119,8	141,3	160	148	13,7	25,7
x.34	117,0	136,8	137	139	19,3	27,8
x.35	128,5	133,5	161	144	19,6	23,5
x.36	114,3	133,0	157	138	14,2	26,1
x.37	126,5	141,5	179	153	17,1	29,9

De forma geral e conforme esperado, o modelo do *leadtime* médio obtém valores de *leadtimes* médios melhores em praticamente todos os experimentos, exceto nos experimentos 7.1, 7.2 e 7.3 onde os valores são iguais aos de 8.1, 8.2 e 8.3, e no experimento 7.31 em que ele é maior do que o 8.31. À medida que a instância do problema cresce, ou seja, a dificuldade imposta ao *solver* aumenta, a diferença entre os valores obtidos para o *leadtime* médio para as soluções encontradas por cada um dos modelos também aumenta. Similarmente, para o modelo que otimiza o *makespan*, os valores encontrados para essa variável de decisão são em geral melhores nesse modelo. A exceção nesse caso são os experimentos x.1 a x.5, já comentados no parágrafo anterior, em que os valores são iguais, e nos experimentos x.15 e x.34 em que os valores de *makespan* obtidos a partir da solução do modelo do *leadtime* médio foram menores que as obtidas no modelo que otimiza o *makespan* diretamente.

Para os experimentos 8.15 e 8.34, em que o modelo *makespan* obteve valores de *makespan* piores que o modelo do *leadtime* médio, os valores do *gap* de otimalidade dessas instâncias foram respectivamente 42,1% e 27,8%. Para esses mesmos casos nota-se que as melhores soluções, ainda que não ótimas, foram encontradas bem próximas do tempo limite de processamento dos problemas de 10.800 segundos ou 3 horas (Tabela 4.6, coluna tempo). Essas duas observações levaram ao entendimento de que para essas instâncias, faltou tempo de processamento para que o *solver* pudesse fechar mais o *gap*, obtendo melhores soluções. Esse mesmo raciocínio aplica-se ao experimento 7.31. Os experimentos 7.31 e 8.34 foram repetidos, executados por 10 horas, a fim de explorar melhor esse ponto observado e os resultados são detalhados na Tabela 4.8.

Tabela 4.8: Resultados dos experimentos 7.31 e 8.34, executados por 10h.

Exp.	Leadtime Médio (períodos)	Makespan		Gap (%)	a_1	a_2	a_3	MOD total	Ocup. Média a_1	Ocup. Média a_2	Ocup. Média a_3	Ocup. Média Total	Tempo (segundos)
		(períodos)	(dias)										
7.31	105,8	148,0	10,6	11,6	5	10	7	22	40%	59%	58%	54%	35.779
8.34	131,3	135,0	9,6	25,7	4	10	6	20	55%	65%	74%	66%	25.505

Conforme é possível notar nos resultados apresentados na Tabela 4.8, para os experimentos 7.31 e 8.34 o *solver* conseguiu obter soluções melhores, conforme esperado. No experimento 7.31 o *leadtime* médio foi reduzido de 111 para 105,8 e o *gap* de otimalidade caiu de 15,8% para 11,6%. Para 8.34 o *makespan* foi de 135, melhor que 137 de seu equivalente 7.34 do modelo do *leadtime* médio, reduzindo o *gap* de 27,8% para 25,6%.

Embora em alguns casos o modelo do *leadtime* tenha chegado a resultados ótimos de *makespan* e vice-versa, isso aconteceu em condições muito específicas de grandes quantidades de força de trabalho (MOD) disponíveis. Isso levou a soluções em que as quatro aeronaves foram praticamente executadas todas em paralelo, o que leva a igualar essas variáveis de decisão, pois elas tendem a ser iguais ao caminho crítico. Ressalta-se então que estes resultados não são uma regra, e isso pode ser notado, por exemplo, nos experimentos x.6, x.7 e x.11, em que ambos os modelos chegaram a soluções ótimas, mas com resultados diferentes de *leadtime* médio e *makespan*.

Nos experimentos 7.26 e 8.26 foi testada a configuração ideal de mão de obra por especialidade ($a_1 = 4$, $a_2 = 12$, $a_3 = 8$). A Tabela 4.9 ilustra os resultados. Para ajudar na comparação, foram inseridos nesta tabela os resultados dos experimentos 7.11 e 8.11, experimentos nos quais a MOD absoluta é a mesma dos experimentos 7.26 e 8.26, para os quais o *solver* encontrou uma solução ótima. Note, que em ambos os casos, a solução com a mão de obra considerada ideal, foi melhor que com a configuração de MOD distribuída por igual ($a_1 = 8$, $a_2 = 8$, $a_3 = 8$), dos experimentos 7.11 e 8.11. O ganho pode ser observado tanto no *leadtime* médio quanto no *makespan* das soluções, sendo que o modelo do *makespan* obteve ganhos mais significativos. Isso mostra que a área de planejamento está em um bom caminho quando calcula a MOD dita ideal para cada manutenção.

Foram testadas ainda diversas variações da distribuição da MOD (experimentos 7.27 a 7.37 e 8.27 a 8.37), com uma redução gradativa da MOD absoluta de até 25%, chegando a um total de 18 técnicos. Nos experimentos 7.27 e 8.27, com uma mudança na distribuição das especialidades, mas mantendo a mesma quantidade absoluta de MOD (24 técnicos), foram

obtidos resultados novamente melhores para o *leadtime* médio e o *makespan* em ambos os modelos. Na configuração de MOD do problema 8.28 e 8.29, foram obtidas soluções muito próximas das obtidas na configuração ideal (experimento 8.26), com 1 pessoa a menos. Com uma redução de 20,8% na MOD absoluta (5 pessoas), em 7.36, o modelo do *leadtime* médio obteve essa variável de decisão com apenas 1 dia a mais que a melhor solução encontrada no problema 7.27 com MOD absoluta de 24 pessoas.

Tabela 4.9: Tabela comparativa de experimentos baseados na MOD ideal.

Exp.	Leadtime Médio (períodos)	Makespan (períodos)	Makespan (dias)	Gap (%)	a_1	a_2	a_3	MOD Total	Ocup Média a_1	Ocup Média a_2	Ocup Média a_3	Ocup Média Total
7.11	103,3	137	9,8	0,0	8	8	8	24	27%	80%	54%	54%
7.26	102,8	129	9,2	11,6	4	12	8	24	57%	57%	58%	57%
7.27	99,5	128	9,1	6,6	6	10	8	24	38%	68%	58%	57%
7.28	103,3	137	9,8	11,1	4	11	8	23	54%	58%	54%	56%
7.29	101,3	137	9,8	7,5	5	10	8	23	43%	64%	54%	56%
7.30	109,0	142	10,1	15,6	4	11	7	22	52%	56%	60%	56%
7.31	111,0	150	10,7	15,8	5	10	7	22	39%	58%	57%	53%
7.32	112,0	133	9,5	16,3	4	10	7	21	55%	66%	64%	63%
7.33	119,8	160	11,4	13,7	3	10	7	20	61%	55%	53%	55%
7.34	117,0	137	9,8	19,3	4	10	6	20	54%	58%	54%	56%
7.35	128,5	161	11,5	19,6	3	10	6	19	61%	54%	62%	58%
7.36	114,3	157	11,2	14,2	4	9	6	19	47%	62%	63%	59%
7.37	126,5	179	12,8	17,1	3	9	6	18	55%	54%	55%	55%
8.11	116,0	126	9,0	0,0	8	8	8	24	29%	87%	59%	58%
8.26	113,3	116	8,3	19,0	4	12	8	24	64%	63%	65%	64%
8.27	109,3	113	8,1	13,3	6	10	8	24	43%	78%	66%	65%
8.28	107,5	117	8,4	20,1	4	11	8	23	63%	68%	64%	66%
8.29	114,3	118	8,4	20,7	5	10	8	23	50%	74%	63%	65%
8.30	116,8	125	8,9	21,3	4	11	7	22	59%	64%	69%	65%
8.31	108,8	120	8,6	19,9	5	10	7	22	49%	73%	71%	67%
8.32	126,3	131	9,4	25,5	4	10	7	21	56%	67%	65%	64%
8.33	141,3	148	10,6	25,7	3	10	7	20	67%	59%	58%	60%
8.34	136,8	139	9,9	27,8	4	10	6	20	54%	64%	73%	64%
8.35	133,5	144	10,3	23,5	3	10	6	19	68%	61%	69%	64%
8.36	133,0	138	9,9	26,1	4	9	6	19	54%	71%	73%	68%
8.37	141,5	153	10,9	29,9	3	9	6	18	64%	63%	65%	64%

É possível observar nas tabelas de resultados a ocupação média calculada da MOD disponível em cada experimento. Pode-se notar nas Tabelas 4.5 e 4.6, que nos experimentos com MOD abundante e igualmente distribuída, tem-se os menores valores de

ocupação média. Conforme esperado, esses valores aumentam à medida que a MOD é restringida e a ocupação média para cada especialidade fica significativamente desbalanceada nesses casos. Nos experimentos x.16 a x.25 vê-se que esse desbalanceamento pode ser melhorado com bons resultados das funções objetivo, mas ainda assim o desbalanceamento permanece. O melhor balanceamento dessa ocupação, e melhor resultado de ocupação total é obtido no experimento x.26, que é feito com a MOD calculada como ideal. Esse equilíbrio, na utilização da MOD, é mantido nos experimentos em que a força de trabalho é reduzida e a proporcionalidade entre os tipos de especialidade (a_1 , a_2 e a_3) do experimento x.26, é mantida. Esses são os experimentos x.26 a x.37.

Isso pode levar a crer que o equilíbrio entre os tipos de força de trabalho é o melhor caminho. Nem sempre a melhor proporção de ocupação entre as especialidades leva aos melhores resultados nas funções objetivo. Nos experimentos 8.35 e 8.36 a MOD absoluta é igual a 19 pessoas. Em 8.35 a ocupação total é 64% e a ocupação entre especialidades está melhor balanceada do que em 8.36. Em 8.36 a ocupação entre especialidades está menos balanceada, no entanto, a ocupação total é 68% e o resultado da função objetivo é melhor do que o de 8.35.

O gráfico da Figura 4.13 compara os valores de *leadtime* médio obtidos pelos dois modelos. À medida que o tamanho do problema aumenta (i.e. MOD é reduzida), os valores dessa variável de decisão também aumentam. O maior valor é atingido nos experimentos x.15, em que a MOD é a mínima estudada. Nesse mesmo ponto, a diferença entre os valores de *leadtime* médio obtidos por cada um dos modelos também é a maior. Fica claro pelo gráfico, conforme anteriormente já comentado, que o modelo (13a)-(18), obtém soluções com melhores valores de *leadtime* médio que o modelo (13b)-(18), conforme esperado.

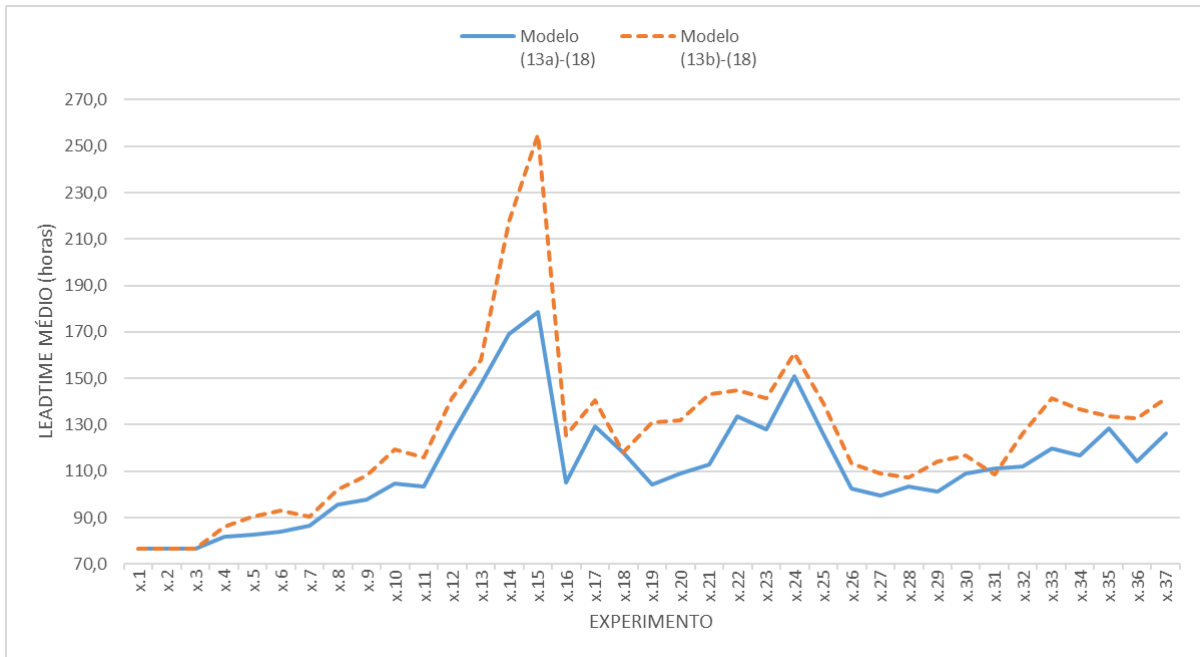


Figura 4.13: Gráfico comparativo, aumento do tamanho do problema versus aumento do *leadtime* médio.

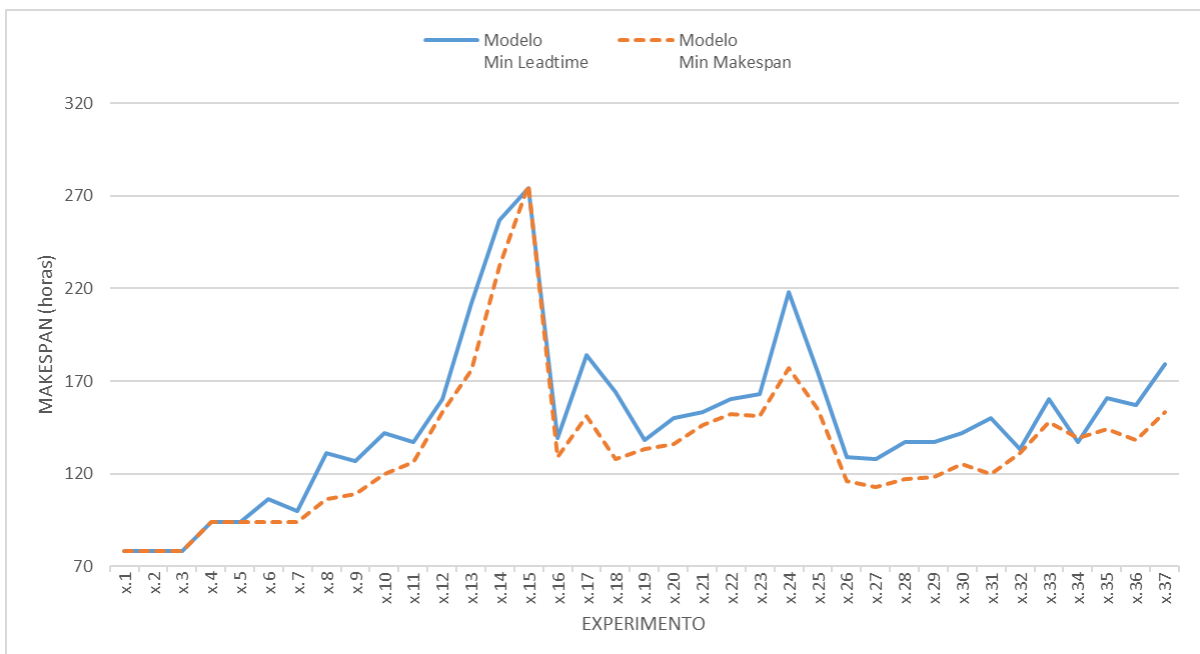


Figura 4.14: Gráfico comparativo, aumento do tamanho do problema versus aumento do *makespan*.

O gráfico da Figura 4.14 compara os valores de *makespan* obtidos pelos dois modelos. É possível notar que a medida que a MOD disponível é reduzida, os valores do *makespan* aumentam significativamente, chegando a um pico de mais de 270 horas para os dois modelos, no experimento x.15 no qual a MOD disponível é de 4 pessoas por especialidade ($a_1 = 4$, $a_2 = 4$, $a_3 = 4$). Outro ponto, também já discutido previamente, mas que fica mais claro

pelo gráfico, é que para a maioria das instâncias o modelo (13b)-(18) que minimiza o *makespan*, obteve soluções melhores nessa variável de decisão que o modelo (13a)-(18) para os problemas de programação sob análise.

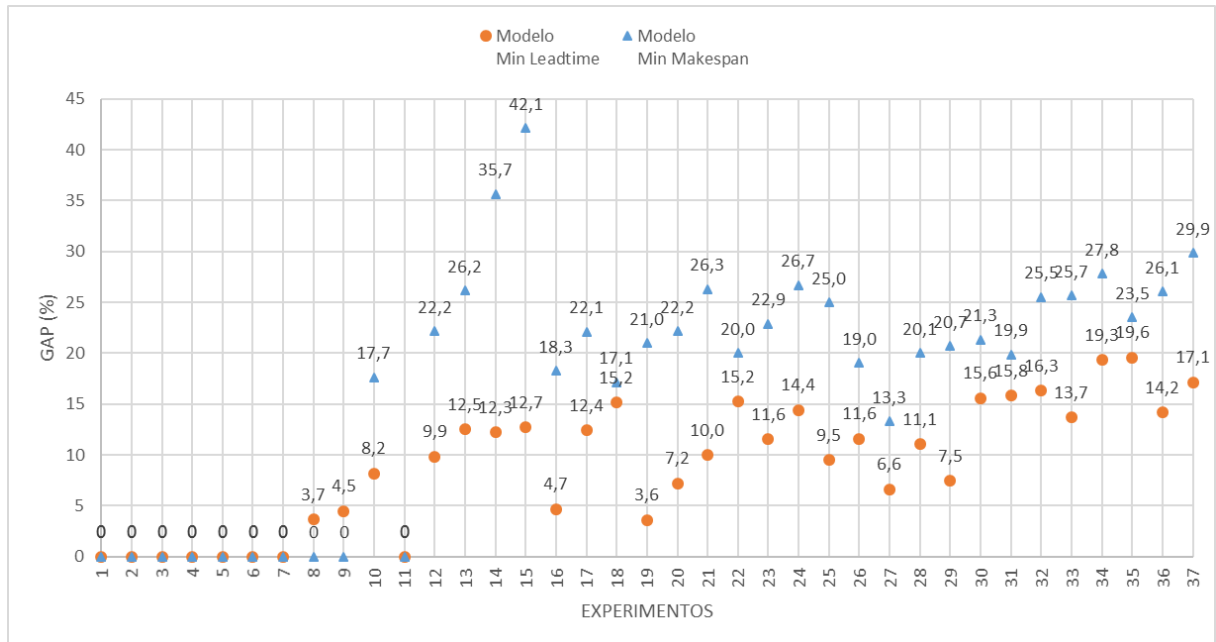


Figura 4.15: Dispersão dos valores dos *gaps* de otimalidade.

Ainda comparando os resultados dos dois modelos, o gráfico de dispersão da Figura 4.15 mostra a evolução do *gap* de otimalidade das soluções à medida que a complexidade do problema aumenta. A conclusão desses resultados é que, para o modelo da função objetivo de minimização do *makespan*, foram obtidas mais soluções com *gap* de otimalidade zero, pois esse modelo possui uma quantidade maior de pontos no valor zero. No entanto, nesse mesmo modelo, para as soluções com *gap* diferente de zero, em todos os casos estes foram significativamente maiores do que os *gaps* obtidos pelo modelo com a função objetivo de minimizar o *leadtime* médio. É importante lembrar, ao analisar esse ponto, que os modelos otimizam variáveis de decisão diferentes, o primeiro o *leadtime* e o segundo o *makespan*. Outro ponto é que o *gap* maior no modelo (18b)-(23), pode significar que o limitante superior utilizado pode ser melhorado.

Outra observação prevista deste estudo está relacionada aos resultados obtidos a partir de cada uma das diferentes funções objetivo dos modelos. Essa observação pode ser mais facilmente visualizada comparando as Figuras 4.16 e 4.17 (páginas 83 e 84), que mostram o diagrama de Gantt da mesma instância do problema dos experimentos 7.7 e 8.7,

respectivamente. As soluções apresentam *makespans* 100 e 94 horas respectivamente. Embora o modelo (13b)-(18), apresente solução com *makespan* melhor, tendência já anteriormente demonstrada, este modelo tende a entregar todo o projeto junto, ou seja, as quatro aeronaves mais próximas do fim do projeto (Figura 4.17). Já no modelo (13a)-(18), a solução tende a comprimir as atividades de cada aeronave, buscando entregar com a maior brevidade as tarefas cada uma destas (Figura 4.16). É como se o modelo (13a)-(18) tratasse cada uma das aeronaves da rede como projetos menores, buscando minimizar o *makespan* desses projetos individualmente, e as primeiras aeronaves a serem iniciadas tendem a serem concluídas mais rapidamente uma vez que foram iniciadas.

Do ponto de vista prático, as duas abordagens podem ser interessantes, e ficou estabelecido um *trade-off*: melhor entregar todo o projeto um pouco mais rápido (mínimo *makespan*) ou levar um pouco mais de tempo para concluir todo o projeto, entregando as aeronaves individualmente mais rápido (mínimo *leadtime* médio)? Para o experimento utilizado como exemplo são mostrados os períodos de entrega de cada uma das aeronaves na Tabela 4.10. As aeronaves são ordenadas por ordem de conclusão, de maneira a permitir a comparação entre a primeira aeronave entregue por cada modelo, a segunda, e assim por diante. É possível notar na Figura 4.19, que a primeira aeronave a chegar, não é, necessariamente, a primeira a ser entregue, principalmente no modelo *makespan*. Deve-se notar na Tabela 4.10 que, apesar da diferença no *makespan* entre as soluções, apenas uma aeronave do experimento 7.7 é entregue após a última do 8.7. O experimento 7.7, entrega, pelo menos 3 aeronaves em tempo significativamente menor se comparado à diferença de tempo entre os *makespans* das soluções.

Tabela 4.10: Período da entrega de cada aeronaves dos experimentos 7.7 e 8.7

Aeronave	Período de entrega	Período de entrega
Entregue	Exp. 7.7	Exp. 8.7
1 ^a	75	82
2 ^a	84	92
3 ^a	88	94
4 ^a	100	94
<i>Leadtime</i> Médio	86,8	90,5
<i>Makespan</i>	100	94

Cabe à empresa, mediante sua estratégia de negócio, decidir a melhor opção no *trade-off* colocado. Cabe a esta ainda, avaliar a oportunidade que se apresenta de entregar o projeto em um tempo um pouco maior, no entanto com uma economia de recurso que pode ser significativa. Os ganhos precisam ser calculados e analisados pela gerência da empresa mediante os impactos dessa possível solução em cada caso.

Foram realizados por fim, dois testes adicionais com os modelos, que buscaram avaliar a possibilidade de priorização das aeronaves dentro do problema. Para isso foram selecionados os experimentos 7.34 e 8.34, nos quais a MOD disponível é equivalente a 75% da mão de obra calculada como ideal ($a_1 = 3$, $a_2 = 9$, $a_3 = 6$). Nesses experimentos o modelo (13a)-(18) teve um *gap* igual a 17% e no modelo (13b)-(18) este foi igual a 29%. Para esse teste foi definido que a aeronave 3 do modelo, a terceira aeronave a chegar, deve ser entregue antes das demais, e para tal foi definido um prazo limite para ela de 85 períodos de tempo ou 85 horas, prazo ligeiramente maior que o necessário para execução de uma aeronave com a MOD ideal. Para isso foi definido que tarefa 49 da rede da Figura 4.7 não deve ser maior que 85 períodos de tempo, ou seja, $d_{49} = 85$. A tarefa 49 é a última tarefa da aeronave 3.

As Figuras 4.18 e 4.19 mostram o gráfico de Gantt para cada um dos experimentos 8.34, com e sem a priorização respectivamente, permitindo comparação. É possível observar na Figura 4.18 que a última tarefa da terceira aeronave ($j = 49$, tarefa circulada) termina junto com as demais, no fim do projeto, no período 153. Já na Figura 4.19 (destacado pelas setas) essa mesma tarefa termina bem antes das demais, no período 84, o que prova que o modelo (13b)-(18) é capaz de priorizar uma aeronave quando definido um período alvo factível. Para o modelo (13a)-(18), essa priorização também foi bem sucedida, no experimento sem a priorização a terceira aeronave foi concluída do período 179, e no experimento com a priorização essa aeronave foi concluída no período 81.

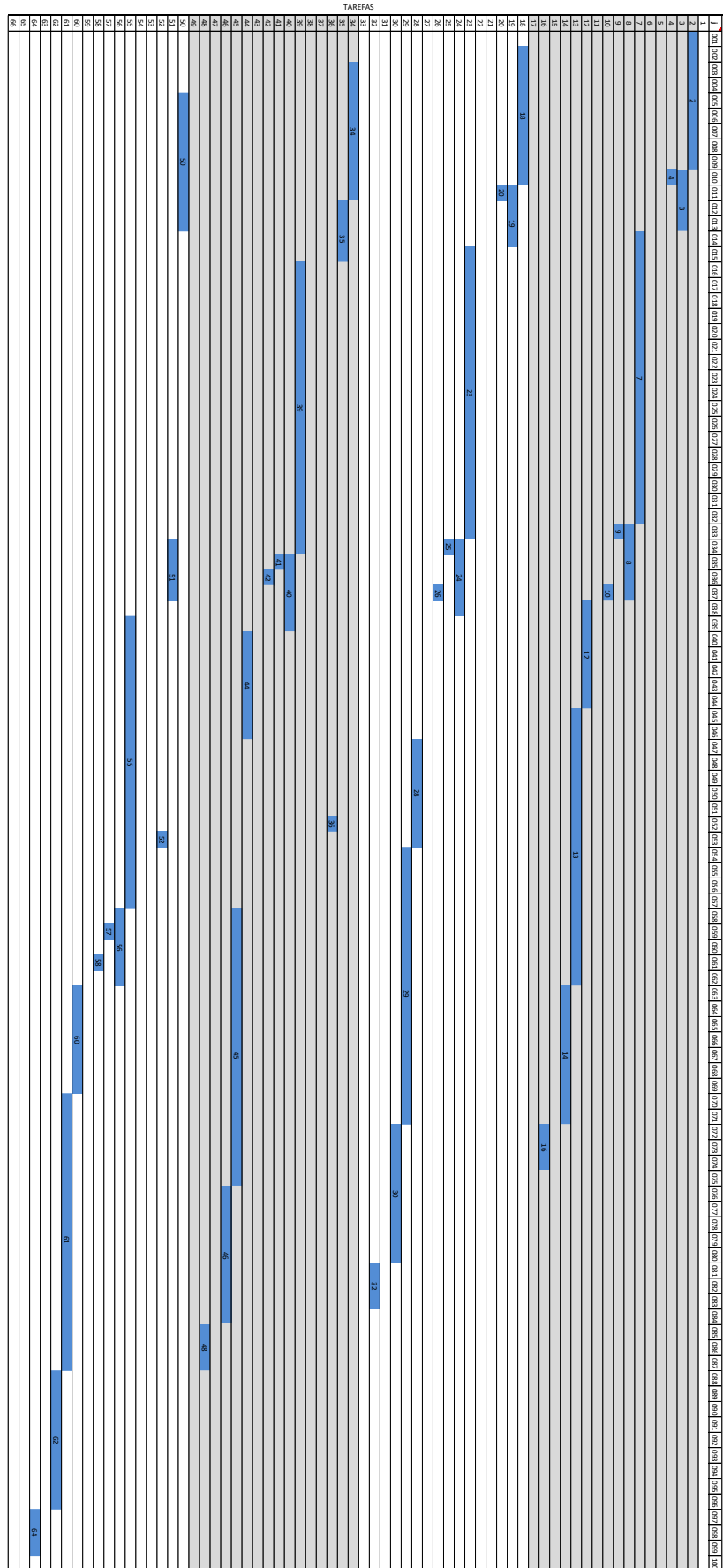


Figura 4. Erro! Indicador não definido.: Gráfico de Gantt da solução do experimento 7.7.

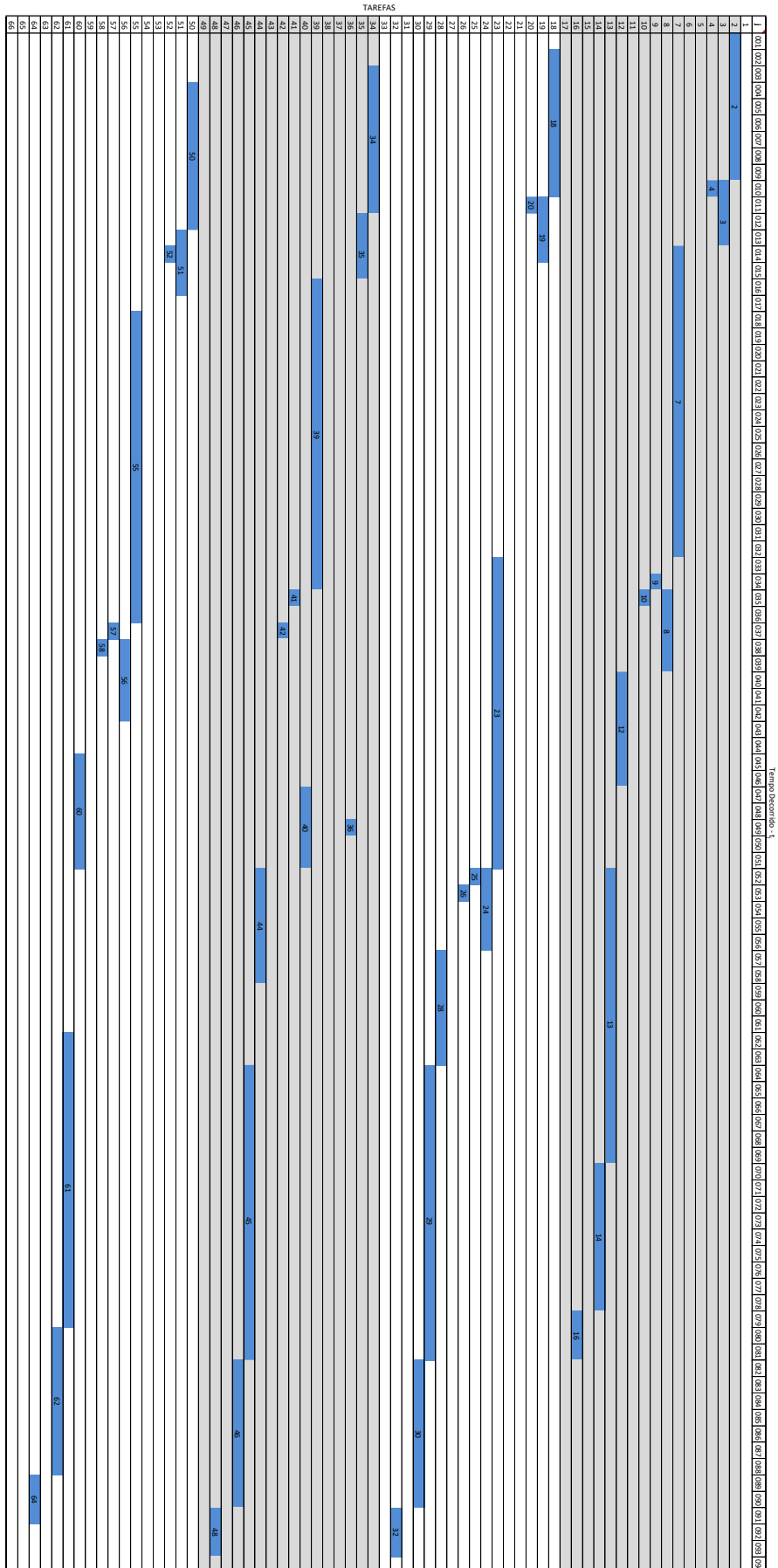


Figura 4.16: Gráfico de Gantt da solução do experimento 8.7.

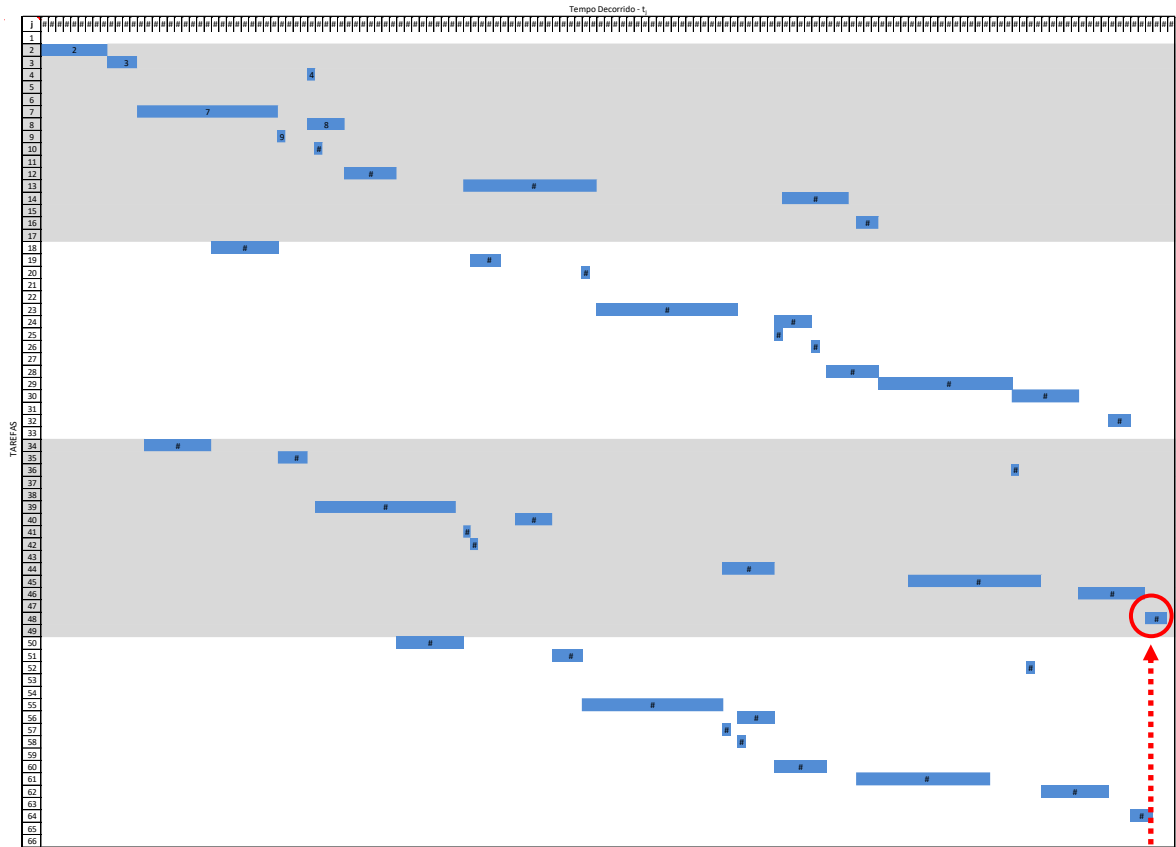


Figura 4.17: Gráfico de Gantt do experimento 8.34 sem a priorização da aeronave 3.

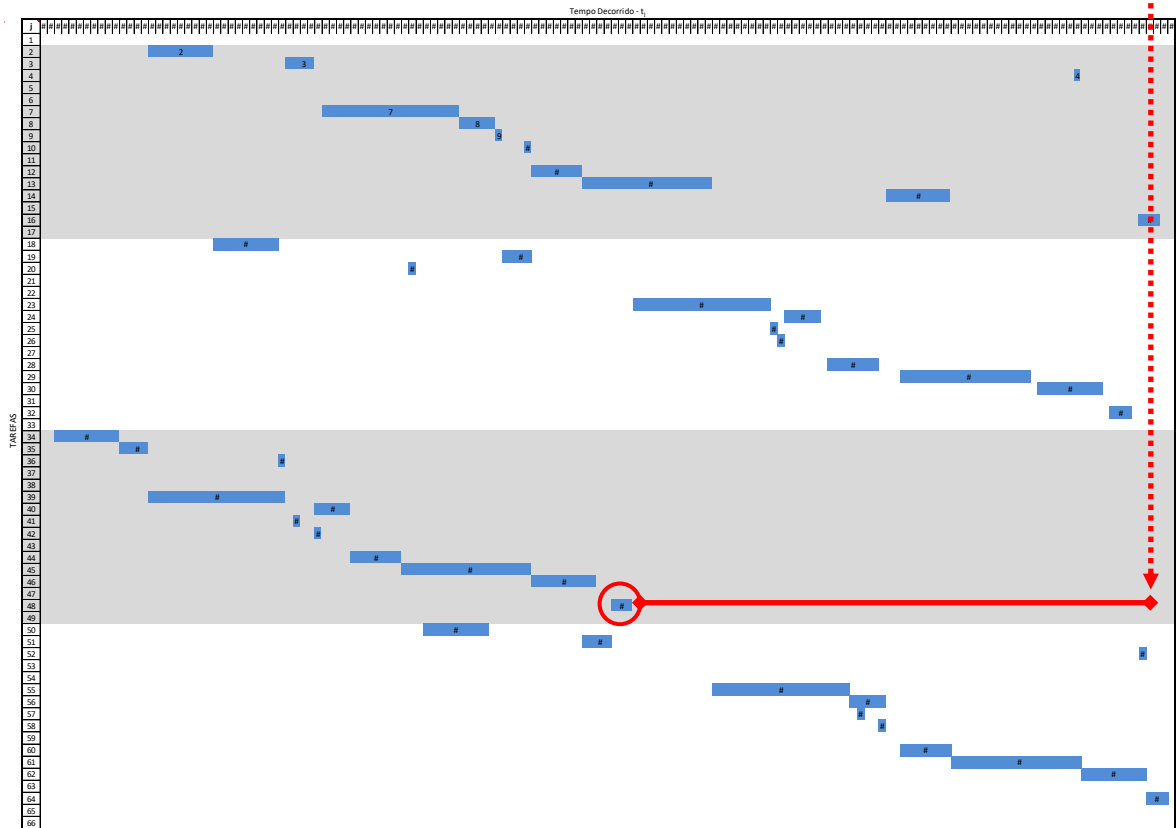


Figura 4.18: Gráfico de Gantt do experimento 8.34 com a priorização da aeronave 3.

5 CONCLUSÕES

O presente trabalho avaliou, por meio de modelos de programação inteira mista, o problema da programação de atividades de manutenção em aeronaves, utilizando dados reais de um estudo de caso. Foram, para tal, propostos dois modelos de otimização, baseados no clássico *Resource Constrained Project Scheduling Problem*. Foram propostas duas abordagens distintas, por meio de funções objetivo diferentes. Uma trabalha com a minimização do *leadtime* médio, ou o tempo de atravessamento total médio de cada aeronave pelo processo de manutenção, enquanto a outra trabalha a minimização do *makespan* do projeto propriamente dito. Ao final, o objetivo principal é o mesmo, minimizar uma determinada função objetivo para o problema, considerando como recurso restrito a força de trabalho especializada das pessoas (MOD), entre outras restrições.

Estudou-se sequenciar a programação das atividades de manutenção de 4 aeronaves distintas, que chegaram para manutenção com um intervalo de 1 período de tempo entre elas. Esse exemplo gerado foi fictício, no entanto, construído sobre dados reais de uma manutenção de pequeno porte, com a colaboração de um importante centro de manutenção. Para a construção do exemplo e seus parâmetros, foram utilizadas as mesmas abordagens do dia a dia da empresa. Foram feitas algumas simplificações e agrupamentos de tarefas, que permitiram tratar problemas de maior tamanho. Tais agrupamentos também foram feitos à luz de estratégias de precedência, utilizadas em diversos centros de manutenção.

Os experimentos mostraram que é possível obter boas soluções para a aplicação prática com os recursos computacionais hoje existentes. O modelo se mostrou como solução alternativa às práticas vigentes de planejamento, ao encontrar soluções factíveis e melhores de programação das tarefas a partir da mesma quantidade de mão de obra. Ao serem colocadas no modelo a distribuição de MOD por especialidade, dita ideal, não foram obtidas soluções ótimas, mas, as obtidas, estão entre as melhores soluções factíveis. Ao serem observadas a ocupação da MOD disponível de acordo com a especialidade, com pequenos ajustes foi possível aprimorar ainda mais a solução. Por exemplo, podem ser obtidas soluções muito satisfatórias do ponto de vista prático, com uma redução na mão de obra requerida de até 17% em termos absolutos.

A perspectiva dos modelos propostos mostrou, conforme esperado, que para o modelo que otimiza o *makespan* diretamente, o *solver* chega a soluções melhores do ponto de vista do *makespan*, chegando a soluções com *gap* de otimalidade zero em mais casos que o

outro modelo. Nos casos em que o *solver* não conseguiu zerar o *gap* para ambos os modelos, os *gaps* do modelo *makespan* são, significativamente, maiores que os *gaps* do modelo *leadtime*.

Uma conclusão esperada é que o modelo *leadtime* tende a comprimir as atividades de cada aeronave, levando a que estas sejam, individualmente, entregues mais rapidamente que no modelo *makespan*. Este último, embora gere soluções com melhor *makespan*, tende a entregar as aeronaves, individualmente, mais próximas do término do projeto. Isso faz com que as aeronaves, em sua maioria, sejam entregues mais rapidamente pelo modelo *leadtime*, pois acabam sendo executadas em um *leadtime* menor.

Se colocada uma situação em que um atraso na linha de manutenção precise ser resolvido, pode ser mais vantajoso, atrasar, no menor número de aeronaves possíveis, e, até mesmo buscar atrasar apenas a última aeronave da linha. Isso pode ser proporcionado pelo modelo *leadtime*. Por outro lado, pode existir o caso de uma programação de manutenção de diversas aeronaves, em que os prazos permitem certa acomodação das entregas, individualmente, situação em que o modelo *makespan* seria mais útil. Dessa forma, cada modelo tem sua utilidade, cabendo à empresa avaliar o que melhor se adequa a sua realidade.

Uma observação adicional é, de que, em ambos os modelos pode-se priorizar as aeronaves desejadas, de forma a atender alguma particularidade ocorrida no processo de manutenção, ou, então, de forma a atender uma dada situação “especial”, que requeira tal priorização.

Concluimos dessa forma que os modelos propostos atingiram o objetivo desse trabalho ao mostrar que estes podem ser alternativas viáveis de aplicação prática na Indústria da Aviação. A abordagem proposta, de simplificação das redes de precedência dos *checks* de manutenção, mostrou-se igualmente viável, permitindo sair do âmbito de trabalho que avalia a manutenção de apenas uma aeronave, podendo olhar um horizonte de tempo maior e propor alternativas de solução para problemas maiores e mais complexos, conseqüentemente, mais relevantes do ponto de vista do negócio.

No âmbito das perspectivas futuras desse trabalho, seria interessante realizar novos experimentos extraindo um problema real ocorrido e registrado, com algumas manutenções um pouco maiores, e implementá-lo de forma a verificar se os modelos são capazes de gerar soluções melhores, aferindo o realmente ocorrido com o proposto pelos modelos. Outro caminho possível, seria aplicar os modelos na prática de manutenção aeronáutica para melhor avaliar os benefícios das suas soluções, em relação às utilizadas pelas empresas.

Existe a possibilidade de aplicar o RCPSP, modelo *makespan*, a um *check* de manutenção não muito grande, de forma a estudar uma configuração de mão de obra que equilibre resultados satisfatórios, com uma melhor ocupação dos recursos. Isso permitiria prescrever uma solução padrão para determinados tipos de manutenção.

Em outra linha de pesquisa, um pouco mais elaborada, se poderia desenvolver e testar métodos exatos e heurísticos para resolver problemas de tamanhos maiores, quando os modelos são mais difíceis de serem resolvidos por *solvers* em tempos computacionais aceitáveis. Por fim, se poderia estender as abordagens de otimização para tratar os problemas com parâmetros incertos, usando técnicas de programação estocástica e otimização robusta, entre outras.

REFERÊNCIAS BIBLIOGRÁFICAS

ADVANCED SCIENCE NEWS. **The Most Efficient Commercial Jet in The World**. 2017. Disponível em: <<https://www.advancedsciencenews.com/efficient-commercial-jet-world/>>. Acesso em: nov/2018.

AHIRE, Sanjay; GREENWOOD, Garrison; GUPTA, Ajay; TERWILLIGER, Mark. Workforce-constrained Preventive Maintenance Scheduling Using Evolution Strategies. **Decision Sciences**, v. 31, n. 4, p. 833-859, 2000.

AIRBUS. **Winglets: a tip-top solution for more efficient aircraft**. Disponível em: <<https://www.airbus.com/newsroom/news/en/2017/02/winglets-a-tip-top-solution-for-more-efficient-aircraft.html>>. Acesso em: nov/2018.

AIRWAY. **Embraer Apresenta o E175 com Wingtips em Lugar de Winglets**. 2014. Disponível em: <<https://airway.uol.com.br/embraer-apresenta-o-e175-com-wingtips-em-lugar-de-winglets/>>. Acesso em: nov/2018.

ALFARES, Hesham K. Aircraft maintenance workforce scheduling A case study. **Journal of Quality in Maintenance Engineering**, v. 5, n. 2, p. 78-89, 1999.

ANAC - AGÊNCIA NACIONAL DE AVIAÇÃO CIVIL. **ANAC Divulga Dados de Tarifas do 1º tri/2018**. Disponível em: <<http://www.anac.gov.br/noticias/anac-divulga-dados-de-tarifas-do-1otri-2018/press-kit.pdf>>. Acesso em: nov/2018.

ARENALES, Marcos; MORABITO, Reinaldo; ARMENTANO, Vinícius; YANASSE, Horacio. **Pesquisa operacional: para cursos de engenharia**. Elsevier Brasil, 2015.

ATA MSG-3 – AIR TRANSPORT ASSOCIATION. **Operator/Manufacturer Scheduled Maintenance Development Revision 2002.1**, Washington, Estados Unidos, 2002.

ATLI, Omer; KAHRAMAN, Cengiz. Aircraft maintenance planning using fuzzy critical path analysis. **International Journal of Computational Intelligence Systems**, v. 5, n. 3, p. 553-567, 2012.

BAKER, Kenneth R. **Introduction to sequencing and scheduling**. New York: John Wiley & Sons, 1974.

BELIËN, Jeroen; CARDOEN, Brecht; DEMEULEMEESTER, Erik. Improving workforce scheduling of aircraft line maintenance at Sabena Technics. *Interfaces*, v. 42, n. 4, p. 352-364, 2012.

BERTSIMAS, Dimitris; GUPTA, Shubham; LULLI, Guglielmo. Dynamic resource allocation: A flexible and tractable modeling framework. **European Journal of Operational Research**, v. 236, n. 1, p. 14-26, 2014.

BOEING. **787 Dreamliner by Design**. Disponível em: <<https://www.boeing.com/commercial/787/by-design/#/advanced-composite-use>>. Acesso em: nov/2018.

BRASIL. ANAC - Agência Nacional de Aviação Civil. **Regulamento Brasileiro da Aviação Civil – RBAC N° 65**. 16 de maio de 2018. 19 páginas.

BRYMAN, Alan. **Research Methods and Organization Studies**. Unwin Hyman, 1989.

BUTCHERS, Rod E.; DAY Paul R.; GOLDIE Andrew P.; MILLER Stephen; MEYER Jeff A.; RYAN David M.; SCOTT, Amanda C.; WALLACE Chris A. Optimized crew scheduling at Air New Zealand. *Interfaces*, 31(1), 30–56. 2001.

CONWAY, Richard Walter; MAXWELL, William L.; MILLER, Louis W. **Theory of scheduling**. Addison-Wesley, 1967.

CPLEX. Version 12. IBM Corporation. Disponível em: <https://www.gams.com/latest/docs/S_CPLEX.html>. Acesso em: julho/2019.

CROTEAU, Elizabeth. **An On-Event Based Model for Resource Constrained Scheduling of Aircraft Heavy Maintenance Tasks**. 2015. 97 f. Masters (Master of Applied Sciences) – Dalhousie University, Halifax, 2015.

DIJKSTRA, Matthis C.; KROON, Leo G.; SOLOMON, Marc; VAN NUNEN, Jo A. E. E.; Van Wassenhove, Luk N. Planning the size and organization of klm’s aircraft maintenance personnel. *Interfaces*, 24(6), 47–58, 1994.

ELMAGHRABY, Salah E.; HERROELEN, Willy S. On the measurement of complexity in activity networks. **European Journal of Operational Research**, v. 5, n. 4, p. 223-234, 1980.

FAA – FEDERAL AVIATION ADMINISTRATION. **Advisory Circular N 120-17A**. Disponível em: <https://www.faa.gov/documentlibrary/media/advisory_circular/ac%20120-17a.pdf>. Acesso em: nov/2018.

FLIGHT GLOBAL. **Embraer Details E-Jet Efficiency Improvements**. 2013. Disponível em: <<https://www.flightglobal.com/news/articles/embraer-details-e-jet-efficiency-improvements-390504/>>. Acesso em: nov/2018.

GULDEMOND, T. A.; HURINK, J. L.; PAULUS, J. J.; SCHUTTEN, J. M. Time-constrained project scheduling. **Journal of Scheduling**, v. 11, n. 2, p. 137-148, 2008.

HARTMANN, Sönke; BRISKORN, Dirk. A survey of variants and extensions of the resource-constrained project scheduling problem. **European Journal of operational research**, v. 207, n. 1, p. 1-14, 2010.

HERROELEN, Willy. Project Scheduling—Theory and Practice. **Production and operations management**, v. 14, n. 4, p. 413-432, 2005.

HERROELEN, Willy; DE REYCK, Bert; DEMEULEMEESTER, Erik. Resource-constrained project scheduling: a survey of recent developments. **Computers & Operations Research**, v. 25, n. 4, p. 279-302, 1998.

JUNQUEIRA, Viviane Souza Vilela; NAGANO, Marcelo Seido; MIYATA, Hugo Hissashi. Procedure structuring for programming aircraft maintenance activities. **Revista de Gestão**, 2018.

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

MARTINS, Roberto A. Abordagens Quantitativa e Qualitativa. In: MIGUEL, P.A.C. et al. **Metodologia de Pesquisa em Engenharia de Produção e Gestão de Operações**. Rio de Janeiro: Elsevier, 2012. p.47-63.

MATTIODA, Daniel D. **Use of Critical Chain Scheduling to Increase Aircraft Availability**. 2002. 110f. Masters (Master of Science in Logistics Management) – Air Force Institute of Technology, School of Engineering and Management, Wright-Patterson Air Force Base, Ohio, 2002.

MORABITO, Reinaldo; PUREZA, Vitoria. Modelagem e Simulação. In: MIGUEL, P.A.C. et al. **Metodologia de Pesquisa em Engenharia de Produção e Gestão de Operações**. Rio de Janeiro: Elsevier, 2012. p.169-196.

MORTON, Thomas; PENTICO, David W. **Heuristic scheduling systems: with applications to production systems and project management**. New York: John Wiley & Sons, 1993.

NAKANO, Davi N. Métodos de Pesquisa Adotados na Engenharia de Produção e Gestão de Operações. In: MIGUEL, P.A.C. et al. **Metodologia de Pesquisa em Engenharia de Produção e Gestão de Operações**. Rio de Janeiro: Elsevier, 2012. p.65-74.

PINEDO, Michael L. **Planning and scheduling in manufacturing and services**. 2ª Edição. New York: Springer, 2005.

PINEDO, Michael L. **Scheduling: theory, algorithms and systems**. 3ª Edição. New York: Springer, 2008.

PRATT & WHITNEY DEPENDABLE ENGINES. **Pratt & Whitney GTF Engine**. Disponível em: < <https://www.pw.utc.com/products-and-services/products/commercial-engines/Pratt-and-Whitney-GTF-Engine/>>. Acesso em: nov/2018.

PRITSKER, A. Alan B.; WATTERS, Lawrence J.; WOLFE, Philip M. Multiproject scheduling with limited resources: A zero-one programming approach. **Management science**, v. 16, n. 1, p. 93-108, 1969.

QUAN, Gang; GREENWOOD Garrison W.; LIU Donglin; HU Sharon. Searching for multiobjective preventive maintenance schedules: Combining preferences with evolutionary algorithms. **European Journal of Operational Research**, v. 177, n. 3, p. 1969-1984, 2007.

ROLLS ROYCE. **Future Products**. Disponível em: <https://www.rolls-royce.com/products-and-services/civil-aerospace/future-products.aspx#>. Acesso em: nov/2018.

SAFAEI, Nima; BANJEVIC, Dragan; JARDINE, Andrew KS. Workforce-constrained maintenance scheduling for military aircraft fleet: a case study. **Annals of Operations Research**, v. 186, n. 1, p. 295-316, 2011.

SAMARANAYAKE, Premaratne; KIRIDENA, Senevi. Aircraft maintenance planning and scheduling: an integrated framework. **Journal of Quality in Maintenance Engineering**, v. 18, n. 4, p. 432-453, 2012.

SILVA, Bruno Jensen Virginio da. **Programação de montagens em gabaritos com restrições de adjacência na indústria aeronáutica**. 2010. 118 f. Dissertação (Mestrado em Engenharia de Produção) – Centro de Ciências Exatas e Tecnologia, Universidade Federal de São Carlos, São Carlos, 2010.

SILVA, Bruno Jensen Virginio da; MORABITO, Reinaldo; YAMASHITA, Denise Sato. Otimização na programação de montagens na indústria aeronáutica. **Gestão & Produção**, v. 21, n. 1, p. 33, 2014a.

SILVA, Bruno Jensen Virginio da; MORABITO, Reinaldo; YAMASHITA, Denise Sato. Production scheduling of assembly fixtures in the aeronautical industry. **Computers & Industrial Engineering**, v. 67, p. 195-203, 2014b.

SILVA, Bruno Jensen Virginio da; MORABITO, Reinaldo; YANASSE, Horacio Hideki. Programação da produção de gabaritos de montagem com restrições de adjacência na indústria aeronáutica. **Produção**, v. 21, n. 4, p. 696-707, 2011.

SILVA, Bruno Jensen; MORABITO, Reinaldo. Planejamento de uma montagem estrutural aeronáutica de jatos utilizando abordagens de programação de projetos com recursos restritos e múltiplos modos. **Gestão & Produção**, 22, 2, 254-266, 2015.

SLACK, Nigel; CHAMBERS, Stuart; JOHNSTON, Robert. **Administração da Produção**. 3ª Edição. São Paulo: Atlas, 2009. 747 p.

SRINIVASAN, Mandyam M.; BEST, William D.; CHANDRASEKARAN, Sridhar. Warner Robins Air Logistics Center streamlines aircraft repair and overhaul. **Interfaces**, v. 37, n. 1, p. 7-21, 2007.

VAN DEN BERGH, Jorne; DE BRUECKER, Philippe; BELIËN, Jeroen; PEETERS, Jonas. Aircraft maintenance operations: state of the art. 2013. **Faculteit Economie En Bedrijfswetenschappen**, v. 34, 2013.

YAMASHITA, Denise Sato; MORABITO, Reinaldo. A note on time/cost tradeoff curve generation for project scheduling with multi-mode resource availability costs. **International Journal of Operations Research**, 5 (4), 429-444, 2009.

YAMASHITA, Denise Sato; MORABITO, Reinaldo. Um algoritmo exato para o problema de programação de projetos com custo de disponibilidade de recursos e múltiplos modos. **Pesquisa Operacional**, v. 27, n. 1, p. 27-49, 2007.