



Universidade Federal de Ouro Preto
Escola de Minas
Departamento de Engenharia de Produção



TALLES FRANCISCO DE BRITO COELHO

OTIMIZAÇÃO DO SEQUENCIAMENTO DE MANUTENÇÃO NA INDÚSTRIA AUTOMOTIVA: UMA ABORDAGEM DO PROBLEMA DE SEQUENCIAMENTO DE PROJETO COM RESTRIÇÕES DE RECURSOS UTILIZANDO SOFTWARES DE CÓDIGO ABERTO

Ouro Preto
2026

TALLES FRANCISCO DE BRITO COELHO

**OTIMIZAÇÃO DO SEQUENCIAMENTO DE
MANUTENÇÃO NA INDÚSTRIA AUTOMOTIVA:
UMA ABORDAGEM DO PROBLEMA DE
SEQUENCIAMENTO DE PROJETO COM
RESTRIÇÕES DE RECURSOS UTILIZANDO
SOFTWARES DE CÓDIGO ABERTO**

Monografia apresentada ao Curso de Engenharia de
Produção da Universidade Federal de Ouro Preto como
parte dos requisitos necessários para a obtenção do grau
em Bacharel em Engenharia de Produção.

Orientador: Prof(a). Clarisse da Silva Vieira Camêlo de
Souza

Ouro Preto
2026

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

C672o Coelho, Talles Francisco de Brito.

Otimização do sequenciamento de manutenção na indústria automotiva [manuscrito]: uma abordagem do problema de sequenciamento de projeto com restrições de recursos utilizando softwares de código aberto. / Talles Francisco de Brito Coelho. - 2026.
40 f.: il.: color., gráf., tab..

Orientadora: Profa. Dra. Clarisse da Silva Vieira Camêlo de Souza.
Monografia (Bacharelado). Universidade Federal de Ouro Preto.
Escola de Minas. Graduação em Engenharia de Produção .

1. Fábricas - Manutenção. 2. Problema de Sequenciamento de Projeto com Restrições de Recursos (RCPSP). 3. Otimização combinatória. 4. Python (Linguagem de programação de computador). 5. Software livre - Google OR-Tools. I. Souza, Clarisse da Silva Vieira Camêlo de. II. Universidade Federal de Ouro Preto. III. Título.

CDU 658.5

Bibliotecário(a) Responsável: Maristela Sanches Lima Mesquita - CRB-1716



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
ESCOLA DE MINAS
DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO,
ADMINISTRAÇÃO E ECON



FOLHA DE APROVAÇÃO

Talles Francisco de Brito Coelho

Otimização do Sequenciamento de Manutenção na Indústria Automotiva: Uma Abordagem do Problema de Sequenciamento de Projeto com Restrições de Recursos Utilizando Softwares de Código Aberto

Monografia apresentada ao Curso de Engenharia de Produção da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Engenheiro de Produção.

Aprovada em 09 de Fevereiro de 2026.

Membros da banca

[Doutora] - Clarisse da Silva Vieira Camêlo de Souza - Orientadora (Universidade Federal de Ouro Preto)

[Doutora] - Irce Fernandes Gomes Guimarães - (Universidade Federal de Ouro Preto)

[Doutor] - Helton Cristiano Gomes - (Universidade Federal de Ouro Preto)

Clarisse da Silva Vieira Camêlo de Souza, orientadora do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 26/02/2026.



Documento assinado eletronicamente por **Clarisse da Silva Vieira Camelo de Souza, PROFESSOR DE MAGISTERIO SUPERIOR**, em 02/03/2026, às 12:49, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **1057417** e o código CRC **247F694F**.

Referência: Caso responda este documento, indicar expressamente o Processo nº 23109.001744/2026-86

SEI nº 1057417

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35402-163
Telefone: 3135591540 - www.ufop.br

Este trabalho é dedicado ao corpo docente da Universidade Federal de Ouro Preto, que desempenhou papel crucial em minha formação acadêmica e profissional, ressaltando a importância da educação pública de qualidade como ferramenta transformadora no Brasil.

Agradecimentos

Em primeiro lugar, agradeço a Deus por me proporcionar viver este momento tão sonhado e à minha família. Só cheguei até aqui graças ao suporte incondicional do meu pai, Heberon, da minha mãe, Maria Ildeir, da minha irmã, Clarice, e das famílias Coelho e Hipólito, que são meu porto seguro.

Agradeço também a Normah Saty, aos meus amigos e aos meus irmãos republicanos, pelo suporte incondicional durante minha graduação.

Por fim, agradeço à Universidade Federal de Ouro Preto e à Escola de Minas, que tornaram este sonho possível, assim como aos meus professores.

"Um vencedor é apenas um perdedor que tentou mais uma vez."

Resumo

O presente trabalho aborda o Problema de Sequenciamento de Projeto com Restrições de Recursos (PSPRR) aplicado ao contexto da manutenção industrial em uma grande empresa do setor automobilístico. Diante da competitividade do setor e da necessidade de garantir a alta disponibilidade dos ativos físicos, o estudo tem como objetivo principal minimizar o tempo total de execução (*makespan*) de um conjunto de atividades de manutenção cíclicas, respeitando as restrições de precedência, capacidade de recursos e janelas de tempo limitadas. A abordagem metodológica caracteriza-se como uma pesquisa aplicada e quantitativa, realizada por meio de um estudo de caso. Para a resolução do problema, utilizou-se a modelagem matemática baseada em Programação Linear Inteira Mista (PLIM), propondo uma adaptação ao modelo de [Vieira e Rola \(2024\)](#) para a inclusão de tempos de preparação (*setup*) dependentes da sequência de execução. A implementação computacional foi desenvolvida utilizando a linguagem de programação Python e a biblioteca de código aberto *Google OR-Tools*, através do *solver* CP-SAT. Os resultados obtidos demonstraram a eficácia da proposta, alcançando uma solução ótima com *makespan* de 135 minutos (inferior ao limite de tempo disponível de 150 minutos) e uma taxa de ocupação de mão de obra superior a 94%. O desempenho computacional destacou-se pelo tempo de processamento de 3,96 segundos e baixo consumo de memória. Conclui-se que a aplicação de Pesquisa Operacional suportada por tecnologias de código aberto apresenta robustez técnica e viabilidade econômica, oferecendo uma alternativa eficiente às soluções comerciais para o suporte à tomada de decisão no planejamento e controle da manutenção.

Palavras-chave: Manutenção Industrial, PSPRR, Otimização, Python, Google OR-Tools.

Abstract

This study addresses the Resource-Constrained Project Scheduling Problem (RCPSP) applied to the context of industrial maintenance within a large automotive company. Given the sector's competitiveness and the necessity to ensure high availability of physical assets, the main objective is to minimize the total execution time (makespan) of a set of cyclical maintenance activities while respecting precedence constraints, resource capacities, and limited time windows. The methodological approach is characterized as applied and quantitative research, conducted through a case study. To solve the problem, mathematical modeling based on Mixed-Integer Linear Programming (MILP) was employed, proposing an adaptation of the [Vieira e Rola \(2024\)](#) model to include sequence-dependent setup times. The computational implementation was developed using the Python programming language and the open-source Google OR-Tools library, utilizing the CP-SAT solver. The results demonstrated the proposal's effectiveness, achieving an optimal solution with a makespan of 135 minutes (within the available 150-minute window) and a labor utilization rate exceeding 94%. The computational performance was notable for its processing time of 3.96 seconds and low memory consumption. It is concluded that the application of Operations Research supported by open-source technologies presents technical robustness and economic feasibility, offering an efficient alternative to commercial solutions for decision-making support in maintenance planning and control.

Keywords: Industrial Maintenance, RCPSP, Optimization, Python, Google OR-Tools.

Lista de figuras

Figura 1 – Construção de um modelo.	4
Figura 2 – Dimensões da complexidade de um modelo de otimização.	5
Figura 3 – Grafo de fluxo de recursos inicial entre as atividades.	20
Figura 4 – Grafo de fluxo de recursos entre as atividades.	23
Figura 5 – Gráfico de Gantt do sequenciamento otimizado com tempos de setup.	24
Figura 6 – Perfil de ocupação da mão de obra (apenas tempos de execução).	25
Figura 7 – Gráfico de Gantt do sequenciamento do cenário alternativo.	27
Figura 8 – Perfil de ocupação da mão de obra para o cenário alternativo.	28

Lista de tabelas

Tabela 1 – Configurações de hardware utilizadas no experimento.	17
Tabela 2 – Parâmetros de entrada do modelo.	18
Tabela 3 – Relações de precedência entre as atividades.	19
Tabela 4 – Alocação de mão de obra e tempo de execução por atividade.	19
Tabela 5 – Resultado ótimo do modelo.	21
Tabela 6 – Taxa de ocupação dos recursos.	25
Tabela 7 – Parâmetros de entrada do cenário alternativo.	26
Tabela 8 – Resultado ótimo do cenário alternativo.	26
Tabela 9 – Taxa de ocupação dos recursos no cenário alternativo.	28
Tabela 10 – Matriz de Setup (Parte 1: Destinos 0 a 18)	37
Tabela 11 – Matriz de Setup (Parte 2: Destinos 19 a 37)	39

Lista de abreviaturas e siglas

DEPRO	Departamento de Engenharia de Produção
UFOP	Universidade Federal de Ouro Preto
CT	Custos Totais
GLPK	<i>GNU Linear Programming Kit</i>
NBR	Norma Brasileira
PI	Programação Inteira
PIM	Programação Inteira Mista
PL	Programação Linear
PLIM	Programação Linear Inteira Mista
PMCSO	Programação Mista com Cone de Segunda Ordem
PNL	Programação Não Linear
PQIM	Programação Quadrática Inteira Mista
PR	Programação por Restrições
PSPRR	Problema de Sequenciamento de Projeto com Restrições de Recursos
RAM	<i>Random Access Memory</i>
RCM	Manutenção Centrada em Confiabilidade
TD	Tempo Discreto
TDD	Desagregada em Tempo Discreto
TPM	Manutenção Produtiva Total

Sumário

1	INTRODUÇÃO	1
1.1	Considerações Iniciais	1
1.2	Objetivos	1
1.2.1	Objetivo Geral	1
1.2.2	Objetivos Específicos	1
1.3	Justificativa	2
1.4	Organização do Trabalho	2
2	REVISÃO BIBLIOGRÁFICA	3
2.1	Contexto da Manutenção na Indústria Automotiva	3
2.1.1	Estratégias e Tipos de Manutenção	3
2.1.2	O Desafio do Agendamento da Manutenção	4
2.2	Modelagem matemática e otimização	4
2.3	Problemas de otimização	5
2.4	Definindo o Problema de Sequenciamento de Projeto com Restrições de Recursos	7
2.5	O Modelo Matemático de Vieira e Rola (2024) para o PSPRR	7
2.6	Soluções Heurísticas e meta-heurísticas	10
2.7	Soluções Exatas	11
2.8	A importância de softwares de código aberto na expansão de ferramentas computacionais	11
2.8.1	Python e <i>solvers</i> para otimização	12
3	METODOLOGIA	14
3.1	O estudo de caso	14
3.2	Coleta e Tratamento de Dados	14
3.2.1	Coleta de dados	14
3.2.2	Compilação e Pré-processamento	15
3.3	Adaptações no Modelo Matemático de Vieira e Rola (2024)	15
3.3.1	Alterações Estruturais	15
3.3.2	Matriz de Setup e Nova Restrição Temporal	15
3.4	Implementação Computacional	16
3.4.1	Coleta de consumo de memória	16
3.4.2	Ambiente Computacional	16
3.5	Análises e Pós-processamento	17

4	RESULTADOS E DISCUSSÃO	18
4.1	Análise dos Dados de Entrada	18
4.2	Desempenho Computacional	21
4.3	Análise do Cronograma Otimizado	22
4.4	Alocação de Recursos	24
4.5	Cenário Alternativo	26
4.5.1	Resultados do Novo Cenário	26
5	CONSIDERAÇÕES FINAIS	29
5.1	Conclusão	29
5.1.1	Desempenho Computacional e Viabilidade de <i>Softwares</i> de Código Aberto	29
5.1.2	Eficiência Operacional	30
5.1.3	Otimização de Recursos e Análise de Cenários	30
5.2	Trabalhos Futuros	30
	Referências	32
	Apêndices	36
	APÊNDICE A – MATRIZ DE SETUP (PARTE 1)	37
	APÊNDICE B – MATRIZ DE SETUP (PARTE 2)	39

1 Introdução

1.1 Considerações Iniciais

A indústria automobilística caracteriza-se historicamente por ser um ambiente de elevada competitividade, onde a eficiência operacional e a continuidade dos processos produtivos são fatores críticos para a sobrevivência e lucratividade do negócio. Neste cenário, a gestão da manutenção passa a assumir um papel estratégico, impactando diretamente indicadores fundamentais para o bom desempenho do processo produtivo (HOLWEG, 2008).

No entanto, a execução eficaz das atividades de manutenção, especialmente as preventivas e programadas, consideram, invariavelmente, restrições de recursos. Em uma planta industrial, a realização dessas tarefas compete por mão de obra especializada, ferramentas e por janelas de tempo limitadas para acontecerem (DUFFUAA; RAOUF; CAMPBELL, 1999). O problema de sequenciar estas atividades para que ocorram da maneira mais eficiente possível, respeitando as limitações de recursos e as precedências técnicas, é conhecido na literatura como o Problema de Sequenciamento de Projeto com Restrições de Recursos (PSPRR) (KOLISCH; PADMAN, 2001).

Atualmente, a complexidade dos problemas de engenharia e a necessidade de flexibilidade e transparência quanto as soluções que se propõe resolver esse tipo de problema abrem uma oportunidade para a adoção de *softwares* de código aberto (*open source*). Estas ferramentas surgem como alternativas robustas às soluções comerciais de alto custo, promovendo a democratização do acesso a tecnologias de otimização de ponta através de uma dinâmica colaborativa (RAYMOND, 2001).

1.2 Objetivos

Esta seção detalha os objetivos geral e específicos traçados para a condução desta pesquisa.

1.2.1 Objetivo Geral

O objetivo principal deste trabalho é obter uma solução ótima para o sequenciamento de atividades de manutenção em uma indústria automotiva, utilizando modelagem matemática e ferramentas computacionais de código aberto.

1.2.2 Objetivos Específicos

Para alcançar o objetivo geral, foram estabelecidos os seguintes objetivos específicos:

- Estudar a viabilidade de soluções de otimização no contexto industrial;
- Experimentar e verificar a eficiência de soluções de código aberto na Pesquisa Operacional;
- Utilizar ferramentas de análise da solução ótima para o modelo matemático utilizado.

1.3 Justificativa

A relevância deste trabalho justifica-se sob três perspectivas principais: acadêmica, técnica e econômica. Do ponto de vista acadêmico, contribui para a literatura ao aplicar um modelo matemático de Programação Linear Inteira Mista (PLIM), adaptado de [Vieira e Rola \(2024\)](#), a um caso real, validando a teoria na prática.

Sob a ótica técnica e econômica, a justificativa apoia-se na validação de ferramentas de código aberto. Diferentemente de soluções comerciais (*solvers* proprietários) que impõem barreiras financeiras, a utilização da linguagem Python em conjunto com o *Google OR-Tools* e as demais bibliotecas utilizadas permite que empresas e pesquisadores desenvolvam soluções de otimização de baixo custo e alta performance. A demonstração de que é possível resolver problemas *NP-Difícil* com ferramentas gratuitas, obtendo alta eficiência no uso de recursos e memória, reforça o potencial de democratização da Pesquisa Operacional na indústria nacional.

1.4 Organização do Trabalho

A estrutura desta monografia foi organizada de modo a apresentar logicamente o desenvolvimento da pesquisa:

- O **Capítulo 1 (Introdução)** apresenta o contexto geral do problema explorado no trabalho, assim como apresenta o contexto industrial estudado na aplicação prática das ferramentas de Pesquisa Operacional utilizadas.
- O **Capítulo 2 (Revisão Bibliográfica)** fundamenta os conceitos de manutenção na indústria automotiva, define o PSPRR e discute as técnicas de modelagem matemática e ferramentas de *software* de código aberto.
- O **Capítulo 3 (Metodologia)** descreve o estudo de caso, o tratamento dos dados, as adaptações realizadas no modelo matemático e a implementação computacional.
- O **Capítulo 4 (Resultados e Discussão)** apresenta a solução ótima encontrada, o cronograma gerado, a análise de alocação de recursos e as métricas de desempenho computacional.
- O **Capítulo 5 (Considerações Finais)** sintetiza as conclusões sobre a viabilidade da abordagem proposta e sugere direcionamentos para trabalhos futuros.

2 Revisão Bibliográfica

2.1 Contexto da Manutenção na Indústria Automotiva

A indústria automotiva caracteriza-se por ser um ambiente de elevada competitividade, onde a eficiência operacional e a continuidade dos processos produtivos são fatores críticos para a sobrevivência do negócio. Neste cenário, a gestão da manutenção deixa de ser uma atividade de apoio para assumir um papel estratégico, impactando diretamente indicadores de disponibilidade, qualidade e custo (HOLWEG, 2008).

Segundo Nakajima (1989), a evolução dos sistemas de produção exigiu uma mudança de paradigma na manutenção, transitando de uma abordagem puramente reativa para modelos preventivos e preditivos focados na confiabilidade dos equipamentos.

2.1.1 Estratégias e Tipos de Manutenção

Para compreender a dinâmica da manutenção no setor automotivo, é necessário definir as abordagens clássicas padronizadas pela norma NBR 5462. Conforme a norma, a manutenção pode ser classificada em: manutenção corretiva, preventiva e preditiva.

A manutenção corretiva se caracteriza por ser executada após a ocorrência de uma pane, destinada a recolocar um item em condições de executar uma função requerida. Embora inevitável em certas situações, é a modalidade mais custosa devido à interrupção não planejada da produção. Já a manutenção preventiva, pode ser definida como uma intervenção que busca prevenir uma falha e tem como principal característica sua periodicidade em intervalos predeterminados ou de acordo com critérios prescritos, destinada a reduzir a probabilidade de falha ou a degradação do funcionamento de um item. Por fim, a manutenção preditiva permite garantir uma qualidade de serviço desejada, com base na aplicação sistemática de técnicas de análise, utilizando-se de meios de supervisão centralizados ou de amostragem, para reduzir ao mínimo a manutenção preventiva e diminuir a manutenção corretiva através da predição de falhas nos equipamentos (TÉCNICAS, 1994).

Na indústria automotiva moderna, busca-se a implementação da Manutenção Produtiva Total (TPM), que visa maximizar a eficiência global dos equipamentos através da eliminação de falhas, quebras e defeitos, envolvendo todos os departamentos da organização (NAKAJIMA, 1989). Em paralelo, a Manutenção Centrada em Confiabilidade (RCM) é utilizada para determinar o que deve ser feito para assegurar que qualquer ativo físico continue a fazer o que seus usuários querem que ele faça no seu contexto operacional (MOUBRAY, 2001).

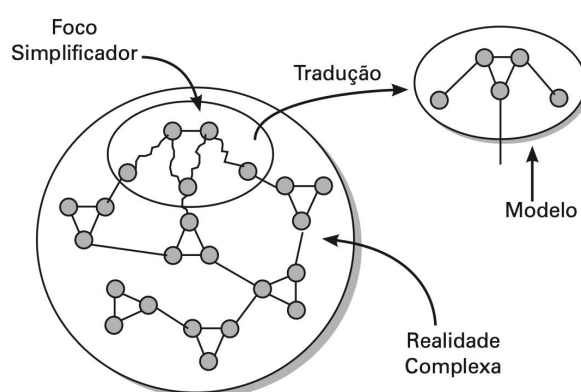
2.1.2 O Desafio do Agendamento da Manutenção

A aplicação dessas estratégias, contudo, esbarra em restrições de recursos. Em uma planta automotiva, a execução das atividades de manutenção preventiva compete por recursos limitados, como mão de obra especializada, ferramentas específicas e janelas de tempo disponíveis entre os turnos de produção. De acordo com [Duffuaa, Raouf e Campbell \(1999\)](#), o problema de agendamento da manutenção envolve determinar a sequência de atividades e a alocação de recursos de forma a minimizar os conflitos com a produção e os custos operacionais.

2.2 Modelagem matemática e otimização

Um modelo pode ser definido como um veículo para uma visão bem estruturada da realidade, sendo assim uma representação simplificada da mesma, que preserva, para determinadas situações e objetivos, uma equivalência adequada, como pode ser observado na figura a seguir:

Figura 1 – Construção de um modelo.

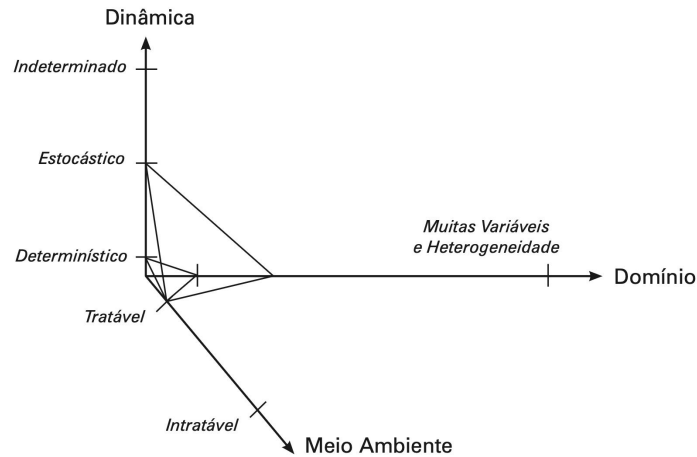


Fonte: [Goldbarg e Luna \(2005\)](#).

Esses modelos são caracterizados por serem definidos de forma lógica e amparada em ferramental matemático. Mais especificamente, um modelo de otimização pode ser definido como uma formulação que busca encontrar uma solução ótima para uma situação do mundo real. Nele são definidos os tipos e nível apropriado de agregação de variáveis, as restrições do problema (quantitativas e lógicas), a natureza dos dados de entrada e de saída, assim como uma função de desempenho, chamada de função objetivo. O termo otimizar é empregado genericamente para contemplar as possibilidades de maximizar ou minimizar a função objetivo.

Sendo assim, os modelos de otimização podem ser aplicados de diferentes formas, de acordo sua dinâmica (como sua estrutura interna se altera ao longo do tempo), meio ambiente (corresponde a quão permeável ele é em relação ao meio ambiente circunvizinho) e seu domínio (homogeneidade de sua estrutura interna, uniformidade de sua morfologia e quantidade de variáveis), como expresso na figura abaixo:

Figura 2 – Dimensões da complexidade de um modelo de otimização.



Fonte: [Goldbarg e Luna \(2005\)](#).

Então, o volume expresso no gráfico da figura acima representa o espaço viável para a atuação dos modelos de otimização, que, quando contínuos, podem ser expressos matematicamente da seguinte maneira:

Minimizar $f(x)$

sujeito a:

$$h_i(x) = 0, \quad i = 1, \dots, m_h$$

$$g_j(x) \leq 0, \quad j = 1, \dots, m_g$$

$$x \in R^n$$

Onde as funções f , h e g são funções contínuas, com domínio em um espaço vetorial com n dimensões (R^n) e contradomínio em R . Em problemas de grande porte, essas funções, em geral, são diferenciáveis. Um problema de otimização discreta é caracterizado por conter soluções viáveis que são conjuntos ou configurações pertencentes a um espaço finito e não vazio ([GOLDBARG; LUNA, 2005](#)).

2.3 Problemas de otimização

Historicamente, classifica-se os problemas de otimização em 8 categorias distintas: Programação por Restrições (PR), Programação Linear (PL), Programação Inteira (PI), Programação Inteira Mista (PIM), Programação Linear Inteira Mista (PLIM), Programação Quadrática Inteira Mista (PQIM), Programação Não Linear (PNL) e Programação Mista com Cone de Segunda Ordem (PMCSO) ([ANAND; AGGARWAL; CHAHAR, 2017](#)).

A Programação por Restrições (PR) é uma técnica utilizada para resolução de problemas de busca combinatória após modelagem do problema. O Problema de Satisfação de Restrições é composto por um conjunto de variáveis, cada uma com um domínio de valores, e um conjunto de restrições que devem ser válidas para subconjuntos dessas variáveis. O objetivo final é encontrar uma atribuição de valores para todas as variáveis que satisfaça simultaneamente todas as restrições, através de métodos como *backtracking* ou *branch and bound*. O processo conhecido como Propagação de Restrições consiste em propagar a informação contida em uma restrição para as restrições vizinhas, a fim de reduzir o espaço de busca a ser explorado (ROSSI; van Beek; WALSH, 2008).

A Programação Linear (PL), para Chen, Batson e Dang (2009), ela consiste no estudo de problemas de otimização com restrições que tem por objetivo encontrar um conjunto de valores para variáveis contínuas que maximiza ou minimiza uma função objetivo linear, enquanto que satisfaz um conjunto de restrições lineares. Além disso, Dantzig (1963), ressalta que a PL pressupõe a proporcionalidade e a aditividade das atividades, permitindo que recursos limitados sejam alocados de forma ótima entre as atividades.

A Programação Inteira (PI), no entanto, é utilizada em situações em que as variáveis de decisão precisam assumir valores discretos. Nesses problemas, as variáveis são restritas a valores inteiros, porém suas funções podem ou não ser lineares (CHEN; BATSON; DANG, 2009).

Portanto, a Programação Inteira Mista (PIM) é frequentemente utilizada em contextos que um cenário de PI apresenta características mistas. Ou seja, um subconjunto de variáveis deve assumir valores inteiros, enquanto o restante pode assumir valores contínuos (CHEN; BATSON; DANG, 2009).

A Programação Linear Inteira Mista (PLIM) se faz uma subcategoria essencial para este campo de estudo. Para o mesmo autor, a PLIM é, portanto, um caso particular de Programação Inteira Mista (PIM), caracterizado por possuir função objetivo e restrições lineares.

Já problemas de Programação Quadrática Inteira Mista (PQIM), de acordo com Lazimy (1982), são aqueles que a função objetivo é quadrática (não linear) e as restrições são lineares nas variáveis inteiras e contínuas, e que as variáveis discretas podem assumir valores não negativos inteiros.

Segundo Fiacco (2013), a Programação Não Linear (PNL) se encontra no espectro mais abrangente da otimização. Para ele, a PNL corresponde a qualquer problema de otimização em que são encontradas relações não lineares na função objetivo ou em suas restrições.

Por fim, os problemas de Programação Mista com Cone de Segunda Ordem (PMCSO) pertencem a classe de problemas de otimização não lineares inteiros mistos e consiste em função objetivo linear que é minimizada sobre a interseção de um subespaço afim e o produto cartesiano de cones de segunda ordem de várias dimensões, com a restrição adicional de que um subconjunto das variáveis deve assumir valores inteiros (DREWES; ULBRICH, 2009).

2.4 Definindo o Problema de Sequenciamento de Projeto com Restrições de Recursos

O Problema de Sequenciamento de Projeto com Restrições de Recursos (PSPRR) pode ser representado através de uma das categorias de problemas listadas na seção 2.3, como a PLIM, por exemplo. Este problema pode ser descrito como a tentativa de sequenciamento de um conjunto de atividades acíclico, ao qual convencionalmente tem-se duas atividades fictícias: uma final e uma inicial. Cada uma dessas atividades contém um determinado e ininterrupto tempo de processamento, assim como relações de precedência. Tais atividades também necessitam de recursos (renováveis ou não renováveis) para serem concluídas (HARTMANN, 2013). Ademais, segundo Kolisch e Padman (2001), o objetivo final da resolução deste clássico problema consiste em encontrar, na maior parte das contribuições acadêmicas, **o menor tempo para a execução de todas as atividades** (*makespan*), dadas suas restrições. Porém, pode ser que sejam utilizadas outras medidas de performance, como a minimização do tempo de fluxo, minimização de atrasos, maximização do valor presente líquido, minimização de custos, dentre outros.

Para Blazewicz, Lenstra e Kan (1983) e Garey e Johnson (1975), problemas de sequenciamento desse tipo apresentam um alto custo computacional para serem resolvidos, e são caracterizados como fortemente *NP-Completo*. Ao analisar as generalizações feitas por estes autores, Artigues, Demassey e Néron (2008) concluem que o PSPRR se caracteriza como *NP-Difícil*. Ainda para estes citados, portanto, esse tipo de problema só pode ser resolvido por meio de soluções exatas em instâncias mais simples. Para problemas maiores, se faz necessária a utilização de métodos que não necessariamente fornecem soluções ótimas, como os métodos heurísticos e meta-heurísticos.

Segundo Koné et al. (2011), a PLIM se apresenta como uma excelente alternativa para resolução de problemas reais que podem ser modelados matematicamente com uma redução de restrições e variáveis a fim de se enquadrar como PSPRR e que apresentam de 25 a 30 atividades. Os mesmos autores propõem comparar os resultados dessas resoluções considerando três métricas: consumo de tempo de processamento, consumo de memória e resultado de suas funções objetivo para um mesmo conjunto de instâncias.

2.5 O Modelo Matemático de Vieira e Rola (2024) para o PSPRR

O trabalho de Vieira e Rola (2024), um exemplo da resolução do PSPRR que utiliza PLIM, consiste em um aprimoramento no modelo matemático proposto por Artigues, Michelon e Reusser (2003), com o objetivo de adicionar parâmetros não previstos pelo modelo original

e deixa-lo mais eficiente, em termos de processamento computacional e tempo para resolução do problema. Portanto, para sequenciar um conjunto de atividades $V = \{1, \dots, n\}$, os autores definiram o conjunto de recursos renováveis $R = \{1, \dots, m\}$ de capacidade finita R_k e o conjunto de recursos não renováveis $N = \{1, \dots, o\}$ representados por N_l os quais elas consomem. O trabalho considera também o conjunto de relações de precedência E entre as atividades i e j e o conjunto de atividades disjuntivas, cuja precedência é desconhecida $\{i, j\} \in D$, onde a atividade j procede a atividade i para ambos. Cada atividade $i \in V$ necessita de uma quantidade não negativa r_{ik} de cada recurso $k \in R$, tem duração p_i e um tempo de *setup* z_i . Para incluir os custos totais do projeto (CT), considera-se o custo associado a utilização dos recursos renováveis C_k , não renováveis C_l e os custos associados ao tempo do projeto (C_p). Ademais, o autor define a variável $S = (S_1, \dots, S_n)$, onde S_i representa o instante de início da atividade i . Tem-se também a variável f_{ijk} , que representa o número de unidades de recurso k diretamente transferido da atividade i para uma atividade j , e a variável x_{ij} , que assume $x_{ij} = 1$ se a atividade j se iniciar após a atividade i ser finalizada e $x_{ij} = 0$ caso contrário.

Vale ressaltar que para este modelo, $s = 0$ e $t = n + 1$ são duas atividade fictícias e que, portanto, $r_{sk} = r_{tk} = R_k, \forall k \in R$ e $p_s = p_t = 0$. Assim sendo, a função objetivo e restrições do modelo podem ser definidas como :

$$\min S_t \quad (1)$$

$$x_{ij} = 1 \quad \forall (i, j) \in E \quad (2)$$

$$x_{ji} = 0 \quad \forall (i, j) \in E \quad (3)$$

$$x_{ij} + x_{ji} = 1 \quad \forall (i, j) \in D \quad (4)$$

$$x_{ij} + x_{ji} \leq 1 \quad \forall (i, j) \notin E, \forall (i, j) \notin D, i \neq j \quad (5)$$

$$s_j - s_i - p_i - z_i - x_{ij} \left(\sum_{i \in V} p_i + \sum_{i \in V} z_i \right) + \left(\sum_{i \in V} p_i + \sum_{i \in V} z_i \right) \geq 0 \quad \forall i \in V \cup \{s\}, \quad \forall j \in V \cup \{t\}, i \neq j \quad (6)$$

$$f_{ijk} - \max\{r_{ik}\} \cdot x_{ij} \leq 0 \quad \forall i \in V \cup \{s\}, \forall j \in V \cup \{t\}, \quad \forall k \in R, i \neq j \quad (7)$$

$$\sum_{j \in V \cup \{t\}} f_{ijk} = r_{ik} \quad \forall i \in V \cup \{s\}, \forall k \in R, i \neq j \quad (8)$$

$$\sum_{j \in V \cup \{s\}} f_{ijk} = r_{jk} \quad \forall j \in V \cup \{t\}, \forall k \in R, i \neq j \quad (9)$$

$$\sum_{i \in V} n_{il} \leq N_l \quad \forall l \in N \quad (10)$$

$$\sum_{l \in N} C_l \cdot N_l + \sum_{k \in R} C_k \cdot R_k + C_p \cdot S_t = CT \quad \forall l \in N, \forall k \in R \quad (11)$$

$$s_i \geq 0 \quad \forall i \in V \cup \{s\} \cup \{t\} \quad (12)$$

$$f_{ijk} \geq 0 \quad \forall i \in V \cup \{s\}, \forall j \in V \cup \{t\}, \quad \forall k \in R \quad (13)$$

$$x_{ij} \in \{0, 1\} \quad \forall i \in V \cup \{s\}, \forall j \in V \cup \{t\} \quad (14)$$

$$CT \geq 0 \quad (15)$$

Note que a equação (1), denominada *função objetivo*, satisfaz a necessidade de minimizar o *makespan* do projeto, haja vista que $p_t = 0$. A restrição (2) garante que as relações de precedência entre as atividades sejam respeitadas. A restrição (3) elimina um circuito entre dois pares de atividades tendo em vista respeitar as relações de precedência entre elas. A restrição (4) trata da relações disjuntivas entre duas atividades, garantindo que duas atividades ($\{i, j\} \in D$), sem prévia relação de precedência declarada, não sejam executadas em paralelo, definindo que deve haver uma ordem de execução entre elas. Já a restrição (5) define que duas atividades sem relação de precedência e sem relação disjuntiva podem ser executadas simultaneamente.

Para melhor entendimento, a equação (6), que garante consistência temporal ao projeto, pode ser reescrita como:

$$s_j \geq s_i + p_i + z_i + x_{ij} \cdot \left(\sum_{i \in V} p_i + \sum_{i \in V} z_i \right) - \left(\sum_{i \in V} p_i + \sum_{i \in V} z_i \right) \quad (16)$$

Perceba que $(\sum_{i \in V} p_i + \sum_{i \in V} z_i)$ representa um número grande o suficiente para viabilizar a comparação lógica nesta restrição. Portanto, se $x_{ij} = 0$, teremos que $s_j \geq s_i + p_i + z_i - (\sum_{i \in V} p_i + \sum_{i \in V} z_i)$, ou seja, a afirmação se torna automaticamente verdadeira. Caso contrário, $s_j \geq s_i + p_i + z_i$.

A restrição (7) limita o fluxo de recurso renovável f_{ijk} da atividade i para a atividade j à quantidade máxima de recurso k disponível, assim como as restrições (8) e (9) dizem respeito às quantidades de recurso necessárias para realização dessas atividades, visto que a lei de conservação do nó estabelece que a quantidade de recurso que entra no nó deve ser igual a quantidade de recurso que sai dele. A restrição (10) delimita a quantidade máxima disponível dos recursos não renováveis n_{il} no projeto. Já as restrições (11) e (15) definem a estrutura de cálculo e o domínio do CT . Por fim, as restrições (12), (13) e (14) definem o domínio das variáveis utilizadas.

2.6 Soluções Heurísticas e meta-heurísticas

Dentre as possibilidades de resolução de problemas do tipo PSPRR, tem-se as soluções heurísticas e meta-heurísticas. De acordo com [Artigues, Demasse e Néron \(2008\)](#), o principal objetivo destas duas técnicas é apresentar uma boa solução em um aceitável período de tempo. Para [Foulds \(1983\)](#), estas soluções não necessariamente são ótimas. Segundo este mesmo autor, os métodos heurísticos podem ser trabalhados em quatro estratégias diferentes. A estratégia da construção consiste na geração incremental de soluções viáveis, com adição gradativa dos componentes que compõem o problema, como propõe o método de [Christofides \(2022\)](#). A estratégia de melhoria consiste no refinamento de uma solução inicial viável de forma iterativa, como propõe a técnica de [Lin e Kernighan \(1973\)](#). A estratégia de análise de componentes utiliza a decomposição do problema em subproblemas independentes, como demonstra [Karp \(1976\)](#). Este autor menciona a estratégia de aprendizado, caracterizada por uso de estruturas de buscas para escolha de qual caminho tomar, através do aprendizado formado a partir de decisões anteriores, como no trabalho de [Krolak, Felts e Marble \(1971\)](#). Por fim, os métodos meta-heurísticos podem ser aplicados como estratégias de busca de alto nível, projetadas para guiar heurísticas subordinadas (como as de construção ou melhoria) de forma a explorar o espaço de soluções de maneira mais eficiente. O objetivo central é escapar de ótimos locais, uma limitação fundamental das heurísticas de melhoria menos sofisticadas ([OSMAN; LAPORTE, 1996](#)).

2.7 Soluções Exatas

Muitos métodos para resolver problemas como o PSPRR são compostos por soluções exatas, que são abordagens que utilizam de diversos algoritmos para solucioná-los de forma ótima, ou seja, encontrando sua melhor solução possível, apesar de seus custos computacionais (DUAN et al., 2023).

Segundo Penha et al. (2017) e Souza e Welerson (2025), o PSPRR já foi trabalhado sob diversas abordagens na literatura científica. Dentre as abordagens para soluções exatas já estudadas, destacam-se trabalhos de grande relevância como a formulação de programação linear 0-1 de Pritsker, Waiters e Wolfe (1969), os métodos de *Branch and Bound* (DEMEULEMEESTER; HERROELEN, 1992; BRUCKER et al., 1998), programação dinâmica (CARRUTHERS; BATTERSBY, 1966) e o algoritmo de Talbot e Patterson (1978).

Na literatura é possível encontrar diversas tentativas de resolução do problema em questão utilizando PLIM. Desses métodos, aqueles em que se utiliza desta técnica para se resolver o problema de forma direta, consistem em duas categorias: aqueles que envolvem um número polinomial de variáveis, conhecido como compacto, e aqueles que envolvem um número pseudo-polinomial de variáveis (KONÉ et al., 2011). Dentre esses, se destacam a formulação básica em tempo discreto (TD) (PRITSKER; WAITERS; WOLFE, 1969), formulação desagregada em tempo discreto (TDD) (CHRISTOFIDES; ALVAREZ-VALDES; TAMARIT, 1987) e a formulação em tempo contínuo baseada em redes de fluxo, presente em Artigues, Michelon e Reusser (2003) e sua adaptação apresentada por Vieira e Rola (2024).

2.8 A importância de softwares de código aberto na expansão de ferramentas computacionais

A crescente complexidade dos problemas de engenharia e a necessidade de reprodutibilidade científica têm impulsionado a adoção de *softwares* de código aberto (*open source*) no ambiente acadêmico e industrial. Segundo Raymond (2001), o modelo de desenvolvimento de código aberto, ao permitir o acesso irrestrito ao código-fonte, fomenta a colaboração descentralizada e a revisão por pares constante, resultando em ferramentas robustas e de rápida evolução.

Diferentemente das soluções comerciais, que muitas vezes impõem barreiras financeiras e de funcionamento interno desconhecido, os softwares de código aberto promovem a democratização do acesso a tecnologias de ponta. Para Hippel (2006), essa democratização permite que usuários deixem de ser apenas consumidores de tecnologia e passem a ser inovadores, adaptando as ferramentas às suas necessidades específicas.

Além da redução de custos com licenciamento, a transparência dos algoritmos em ferramentas *open source* é fundamental para a validação científica. Em otimização, isso garante que

os resultados obtidos possam ser auditados e replicados por outros pesquisadores, fortalecendo a confiabilidade dos estudos (DUAN et al., 2023). Esse ecossistema colaborativo serviu de base para a consolidação de linguagens e bibliotecas que hoje são padrões de mercado.

2.8.1 Python e *solvers* para otimização

Sejam métodos exatos ou heurísticos, a utilização de softwares para resolver o PSPRR é amplamente reconhecida academicamente. Dentre essas ferramentas, destacam-se aqueles baseados em Python, que é uma linguagem de programação que se caracteriza por sua versatilidade e legibilidade. Embora inclua muitos recursos nativos, ao longo dos anos, sua comunidade expandiu sua gama de aplicações através do desenvolvimento de uma grande quantidade de pacotes externos. Dentre esses, o *pandas* e o *numpy* se destacam por fornecer estruturas e métodos para computação de dados de alto desempenho (MUNISAMY, 2025). Esta linguagem contempla também ferramentas para visualização dos dados, como a *matplotlib*, que se faz capaz de produzir gráficos que exigem controle detalhado e visualizações altamente personalizadas (TEAM, 2025), que pode ser combinada com a *networkx*, para plotagem de grafos. Ademais, surgiram diversos pacotes e ferramentas com aplicações matemáticas, incluindo *solvers* para otimização de problemas que utilizam PIM (SANTOS; TOFFOLO, 2020; HAO; HO, 2019).

Para resolução desses problemas via ferramentas de *software*, utiliza-se de Programação Matemática, que é uma linguagem formal utilizada com o intuito de descrever problemas de otimização. Uma vez que um problema é modelado utilizando esta técnica, pode-se então realizar a adaptação deste modelo para uma linguagem de programação, como o Python, e utilizar *softwares* desta mesma linguagem chamados de *solvers* para solucioná-lo. Estes softwares usualmente incorporam uma grande quantidade de algoritmos (com destaque para o método Simplex, Método de Pontos Interiores e o algoritmo de Branch and Bound) para problemas com diferentes características, equilibrando o *trade-off* entre eficiência e abrangência de uso (IOMMAZZO et al., 2020). Esses *softwares* se dividem entre soluções comerciais, como o CPLEX, por exemplo, e soluções de código aberto, como o GLPK e o CP-SAT.

Dentre as soluções comerciais, o CPLEX da IBM se destaca como um dos mais completos e aceitos *solvers* do mercado, pois se trata de um pacote que oferece recursos para resolução através do uso de heurísticas, *lazy constraints*, algoritmo de *Branch and Cuts* para otimização distribuída ou em paralelo, técnica de decomposição de Benders, entre outras funcionalidades. Já o GLPK (GNU *Linear Programming Kit*) é um projeto que foi desenvolvido para resolver problemas de PL de larga escala utilizando os métodos Simplex Primal e Dual e o método de Pontos Interiores (ANAND; AGGARWAL; CHAHAR, 2017; GEARHART et al., 2013). Por fim, o CP-SAT, desenvolvido pela Google, utiliza uma arquitetura chamada de *Lazy Clause Generation* para integrar as técnicas de PR com SAT (Satisfatibilidade), ao mesmo tempo em que incorpora o *Simplex* a fim de obter vantagens (como o relaxamento linear, por exemplo) ao lidar com a parte linear dos modelos (PERRON; DIDIER; GAY, 2023).

Mais especificamente, a arquitetura de resolução de problemas de *solvers* como o CP-SAT

3 Metodologia

O presente capítulo descreve o método de pesquisa utilizado para abordar o PSPRR aplicado ao sequenciamento de atividades de manutenção. Trata-se de uma pesquisa aplicada, de natureza quantitativa, realizada por meio de um estudo de caso em uma grande empresa do setor automobilístico.

A abordagem metodológica fundamenta-se nas técnicas de modelagem matemática e otimização apresentadas nas seções 2.2 e 2.3. Para tanto, a metodologia do trabalho foi dividida nas etapas de definição do estudo de caso, coleta e tratamento de dados, adaptação do modelo matemático, implementação computacional e Análises e Pós-processamento.

3.1 O estudo de caso

O presente estudo trata-se da aplicação de ferramentas de Pesquisa Operacional para sequenciamento de atividades de manutenção em uma planta industrial de uma grande empresa do setor automobilístico, tarefa que hoje é desempenhada sem auxílio de uma ferramenta de otimização. Tais atividades são programadas (cíclicas ou pontuais), ocorrendo majoritariamente em momentos estratégicos de finais de semana. A escolha desses horários visa excluir ou minimizar as perdas advindas de paradas na linha de produção, que são consequências diretas da execução destas manutenções.

Tendo em vista o cenário delineado, tem-se que o problema caracteriza-se, portanto, como um PSPRR, onde se busca obter o menor tempo para execução de todas as atividades (*makespan*), respeitando as limitações de recursos e precedências.

3.2 Coleta e Tratamento de Dados

A etapa de coleta de dados foi realizada mediante análise documental e observação direta, dividida em três fases principais: levantamento de parâmetros, análise documental e cronometragem de tempos de *setup*. Após o final do período de coleta, os dados foram compilados e pré-processados.

3.2.1 Coleta de dados

Na etapa de levantamento de dados, as informações referentes à mão de obra disponível, tempo limite para conclusão do projeto, relações de precedência entre tarefas, duração estimada e quantidade de mão de obra requerida foram obtidas através de interface direta com a equipe de planejamento de atividades da empresa.

Posteriormente, na análise documental, os dados foram extraídos de bancos de dados e planilhas utilizadas pela equipe na semana anterior ao final de semana objeto de estudo.

Por fim, para a construção da matriz de tempos de preparação e deslocamento (*setup* as atividades), foi necessário realizar a cronometragem do tempo médio de deslocamento entre os locais das atividades e o ponto de partida (sala de manutenção).

3.2.2 Compilação e Pré-processamento

Após a coleta, os dados foram compilados e padronizados em planilha eletrônica. Posteriormente, utilizou-se a linguagem de programação Python para a leitura e transformação desses dados. Nesta etapa de pré-processamento, as informações tabulares foram convertidas em estruturas de dados nativas do Python, como dicionários, listas e matrizes, permitindo sua manipulação. Assim como citado na subseção 2.8.1, o uso da biblioteca *pandas* se faz essencial neste processo.

3.3 Adaptações no Modelo Matemático de Vieira e Rola (2024)

Tendo em vista a natureza do problema e características das variáveis, optou-se pela aplicação de PLIM, tendo como referência o modelo proposto por [Vieira e Rola \(2024\)](#), explicado na seção 2.5.

3.3.1 Alterações Estruturais

Devido às características do processo de manutenção estudado, não foram utilizadas relações disjuntivas. Consequentemente, as restrições (4) e (5) foram excluídas. Além disso, foram desconsideradas também as restrições (11) e (15), haja visto que o foco do estudo recai sobre a eficiência da aplicação de ferramentas de código aberto, e não nos impactos financeiros referentes a atividade de manutenção da empresa estudada. Se faz necessário citar também que o estudo limitou-se aos recursos renováveis (mão de obra), desconsiderando a restrição (10).

3.3.2 Matriz de Setup e Nova Restrição Temporal

Devido à natureza mutável dos tempos de deslocamento na planta entre uma atividade e outra, e entre uma atividade e a sala de manutenção (ponto inicial), o parâmetro z_i (tempo de *setup* fixo das atividades) foi substituído por um vetor de dois índices z_{ij} , representando o tempo de setup entre a atividade i e a atividade j . Consequentemente, a restrição (6) foi reformulada para:

$$s_j - s_i - p_i - z_{ij} + x_{ij} \cdot M - M \geq 0 \quad (17)$$

Sendo M :

$$M = \sum_{i \in V} p_i + \sum_{i \in V} \left(\max_{\substack{j \in V \cup \{t\} \\ j \neq i}} \{z_{ij}\} \right) \quad (18)$$

Essa adaptação assegura que, se a atividade j for realizada após a atividade i ($x_{ij} = 1$), o tempo de início de j deve ser maior ou igual ao término de i , mais o tempo de deslocamento z_{ij} . O novo método modifica também a construção do termo M , que passa a calcular o somatório dos valores máximos de j para cada i , de forma a garantir que o mesmo termo seja grande o suficiente para a comparação lógica empregada.

3.4 Implementação Computacional

A implementação computacional do modelo foi realizada utilizando a linguagem *Python* (versão 3.11.11), escolhida por sua versatilidade na manipulação de estruturas de dados e visualização de resultados. Para a resolução do problema de otimização, utilizou-se o pacote de ferramentas *Google OR-Tools* (versão 9.14.6206), através do solver *CP-SAT*¹. Portanto, o algoritmo desenvolvido recebe as estruturas de dados processadas e retorna a programação ótima das atividades, definindo os instantes de início que minimizam o tempo total do projeto.

3.4.1 Coleta de consumo de memória

Assim como mencionado na seção 2.4, para [Koné et al. \(2011\)](#), tem-se que a quantidade alocada de memória RAM está entre as três principais métricas de desempenho de modelos de otimização. Ao mesmo tempo, o solver *CP-SAT* não apresenta este importante indicador em sua versão utilizada neste projeto. A fim de contornar este obstáculo para mensuração dos resultados, a biblioteca *psutil* foi utilizada. Este pacote externo tem como função obter informações sobre a quantidade de memória RAM que o processo está ocupando em um determinado instante. Portanto, ele foi utilizado para a aferição da diferença de consumo de memória entre o início e o fim da execução do modelo.

3.4.2 Ambiente Computacional

Com o objetivo de garantir a reprodutibilidade do estudo de caso e estabelecer uma base de comparação para o desempenho do modelo proposto, descrevem-se a seguir as especificações de *hardware* utilizadas na execução do código:

¹ A escolha da biblioteca de código aberto *CP-SAT* justifica-se por sua arquitetura avançada, conforme mencionado na subseção 2.8.1.

Tabela 1 – Configurações de hardware utilizadas no experimento.

Componente	Especificação
Nome do Modelo	MacBook Air
Identificador do Modelo	Mac14,15
Número do Modelo	MQKT3LL/A
Processador (Chip)	Apple M2
Núcleos de Processamento	8 uni.
Memória RAM	8 GB

Fonte: O autor (2026).

3.5 Análises e Pós-processamento

Após obtenção dos resultados do modelo, a análise dos dados obtidos tem carácter fundamental para determinação do sucesso do projeto. Portanto, utilizou-se de algoritmos de programação para extrair os valores de variáveis chaves a fim de gerar visualizações que fossem úteis para medição da performance do projeto e no processo de tomada de decisão por parte da equipe de programação de manutenções. Para isso, utilizou-se de bibliotecas como *pandas*, *matplotlib* e *numpy* para obtenção de tabelas, gráficos e grafos. Por fim, foi construído um cenário alternativo, com o objetivo de demonstrar a flexibilidade do projeto em testar diferentes condições e hipóteses.

4 Resultados e Discussão

Neste capítulo são apresentados os resultados obtidos a partir da implementação computacional do modelo matemático de otimização adaptado de [Vieira e Rola \(2024\)](#) na seção 3.3, assim como análises posteriores. A análise divide-se em cinco partes principais: Análise das informações de entrada, o desempenho computacional do *solver* no ambiente de *hardware* descrito, a análise do cronograma otimizado (*makespan*), a avaliação da alocação de recursos com a consideração dos tempos de preparação (*setup*) e a construção de um cenário alternativo para o projeto.

4.1 Análise dos Dados de Entrada

O presente estudo de caso foi desenvolvido em uma grande empresa do setor automobilístico, que se divide em múltiplos setores operacionais. Dentre estes, a pesquisa delimita-se à equipe de manutenção do Setor 1, focando na programação de suas atividades programadas (cíclicas ou pontuais), conforme as justificativas estratégicas detalhadas na seção 2.1.

Esta equipe é responsável pela execução de tarefas que devem ocorrer dentro de uma restrita janela de oportunidade ("intervalo de parada"), cujas razões técnicas foram explicitadas na seção 3.1. Especificamente para o cenário analisado, dispõe-se de um intervalo de tempo de 150 minutos (2,5 horas) para a conclusão das ordens de serviço.

Após a aplicação dos procedimentos de coleta descritos na seção 3.2, os dados foram estruturados para alimentar o modelo matemático. A Tabela 2 resume os parâmetros gerais do problema, evidenciando que as 36 atividades mapeadas estão distribuídas ao longo de 3 linhas de produção, que englobam um total de 9 operações distintas. Para o cumprimento desta demanda, a equipe conta com 10 colaboradores, divididos entre 7 mecânicos e 3 eletricitistas.

Tabela 2 – Parâmetros de entrada do modelo.

Parâmetro	Quantidade
Linhas de produção	3
Operações	9
Tempo disponível	150 min
Atividades	36
Mecânicos	7
Eletricistas	3

Fonte: O autor (2026).

Dentre as atividades obtidas acima, tem-se atividades de ajustes mecânicos e preventivas de limpeza em equipamentos e seus respectivos sistemas, assim como lubrificação e substituição

de componentes.

Nesta etapa, foram mapeadas também as relações de dependência entre as tarefas. A Tabela 3 apresenta a estrutura de precedências, indicando quais atividades devem ser obrigatoriamente concluídas antes do início de suas sucessoras:

Tabela 3 – Relações de precedência entre as atividades.

Ativ.	Dep.	Ativ.	Dep.	Ativ.	Dep.
1	–	13	12	25	24
2	–	14	13	26	–
3	2	15	14	27	26
4	3	16	15	28	27
5	4	17	16	29	28
6	5	18	–	30	–
7	6	19	18	31	30
8	7	20	19	32	–
9	8	21	20	33	–
10	–	22	21	34	–
11	10	23	22	35	–
12	11	24	–	36	–

Fonte: O autor (2026).

Foram obtidos também os tempos de execução estimados pela equipe de manutenção, assim como as necessidades de recursos de cada atividade, presentes na Tabela 4, apresentada abaixo:

Tabela 4 – Alocação de mão de obra e tempo de execução por atividade.

Ativ.	Recursos	Tempo (min)	Ativ.	Recursos	Tempo (min)	Ativ.	Recursos	Tempo (min)
1	1 Mecânico	60	13	1 Mecânico	5	25	2 Mecânicos	10
2	1 Mecânico	5	14	1 Mecânico	3	26	2 Mecânicos	20
3	1 Mecânico	10	15	1 Mecânico	5	27	2 Mecânicos	20
4	1 Mecânico	10	16	1 Mecânico	2	28	2 Mecânicos	20
5	1 Mecânico	5	17	1 Mecânico	5	29	2 Mecânicos	20
6	1 Mecânico	3	18	1 Mecânico	20	30	2 Mecânicos	15
7	1 Mecânico	5	19	1 Mecânico	20	31	2 Mecânicos	15
8	1 Mecânico	2	20	1 Mecânico	30	32	1 Mec. + 1 Elet.	60
9	1 Mecânico	5	21	1 Mecânico	30	33	1 Mec. + 1 Elet.	60
10	1 Mecânico	5	22	1 Mecânico	5	34	1 Eletricista	30
11	1 Mecânico	10	23	1 Mecânico	5	35	1 Eletricista	30
12	1 Mecânico	10	24	2 Mecânicos	10	36	1 Mec. + 1 Elet.	120

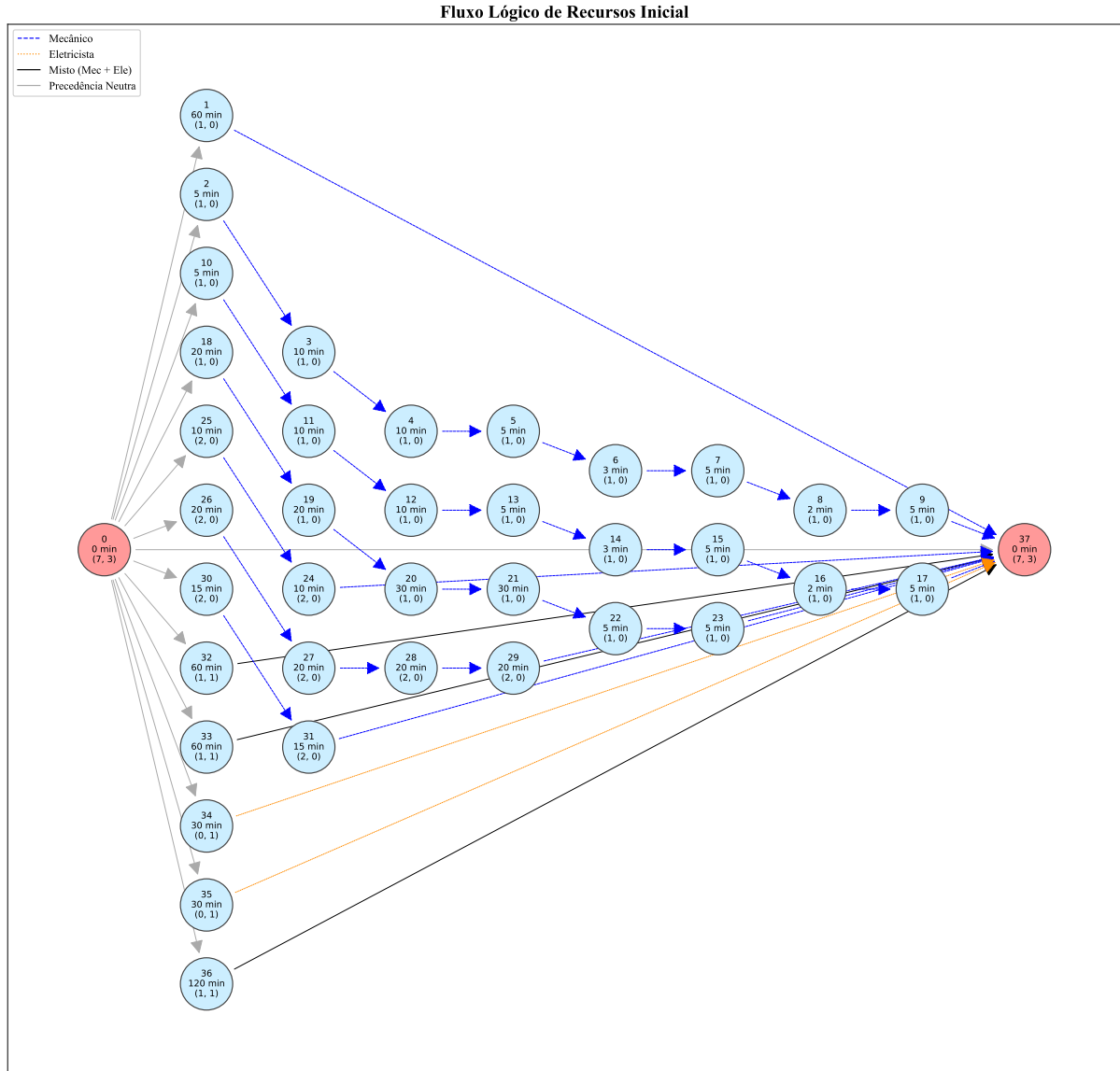
Fonte: O autor (2026).

Note que foram utilizadas as abreviações "Mec." para Mecânico e "Elet." para Eletricista, de forma a facilitar a disposição dessas informações.

Portanto, com o intuito de que as relações entre as atividades e os recursos sejam entendidas de forma clara, utilizou-se das bibliotecas *networkx* e *matplotlib* para exposição das relações

iniciais entre as atividades do processo, assim como as necessidades de recursos, através do grafo da Figura 3, de acordo com os dados obtidos acima:

Figura 3 – Grafo de fluxo de recursos inicial entre as atividades.



Fonte: O autor (2026).

Na figura acima, observa-se que os nós (círculo em azul claro para atividades reais e em vermelho para atividades fictícias) da Figura 3 representam as atividades. Cada um deles exibe o identificador numérico da tarefa, o tempo estimado de execução e, entre parênteses, a demanda de recursos (quantidade de mecânicos e eletricitas, respectivamente). É possível observar também que foram utilizadas setas para representar a precedência entre as atividades, com diferentes estilos de linhas para simbolizar cada possibilidade de necessidade de recursos. A linha pontilhada de cor azul representa a transferência de recursos do tipo "Mecânico", exclusivamente. Já a linha pontilhada laranja representa o fluxo de recursos do tipo "Eletricista", também de forma exclusiva.

Por fim, a linha contínua de cor preta representa o fluxo entre atividades que envolve ambos os tipos citados acima e a linha contínua em cinza representa as relações implícitas nos dados apresentados, denominadas pelo autor deste trabalho de "Precedência Neutra". As relações de Precedência Neutra aparecem entre a atividade fictícia inicial e suas possíveis sucessoras.

Por fim, para capturar a realidade logística da planta industrial, os tempos de deslocamento entre os locais de execução foram cronometrados. Estes dados compõem a matriz de tempos de *setup*, que, devido à sua extensão, os dados são apresentados nos Apêndices A e B, divididos em duas partes, com o objetivo de garantir a legibilidade. Os valores representam o tempo em minutos e os índices representam as atividades. Esta matriz considera tanto o deslocamento entre diferentes operações quanto o trajeto inicial a partir da sala de manutenção. Vale ressaltar que o tempo de preparação para ler as instruções da atividade e coleta de equipamento já está incluso no tempo de execução da atividade.

4.2 Desempenho Computacional

O modelo matemático, adaptado de [Vieira e Rola \(2024\)](#) na seção 3.3 e implementado em Python utilizando a biblioteca *Google OR-Tools*, assim como foi detalhado na seção 3.4, foi submetido aos dados apresentados na seção anterior (4.1), incluindo 2 atividades fictícias (início e fim), totalizando 38 atividades.

A execução ocorreu no ambiente computacional especificado na Tabela 1 do Capítulo 3. O parâmetro de tempo máximo de busca foi configurado para 1.800 segundos (30 minutos), porém, o *solver* CP-SAT demonstrou alta eficiência na busca por uma solução.

Foi encontrada uma Solução Ótima, o que significa que não há possibilidade matemática de existir um cronograma com duração menor do que o encontrado, respeitando-se todas as restrições de precedência, recursos e tempos de deslocamento. Os indicadores de performance computacional obtidos foram:

Tabela 5 – Resultado ótimo do modelo.

Indicador	Valor
Status da solução	Ótima
makespan	135 min
Tempo de Processamento (<i>Wall time</i>)	3,96 s
Uso de memória*	57,59 MB
ramificações	2363

Fonte: O autor (2026).

Na Tabela 5, nota-se que o CP-SAT foi capaz de encontrar a solução ótima de 135 minutos de tempo de projeto. Este resultado incorpora não apenas os tempos de processamento das atividades (p_i), mas também os tempos de deslocamento variáveis dependentes da sequência

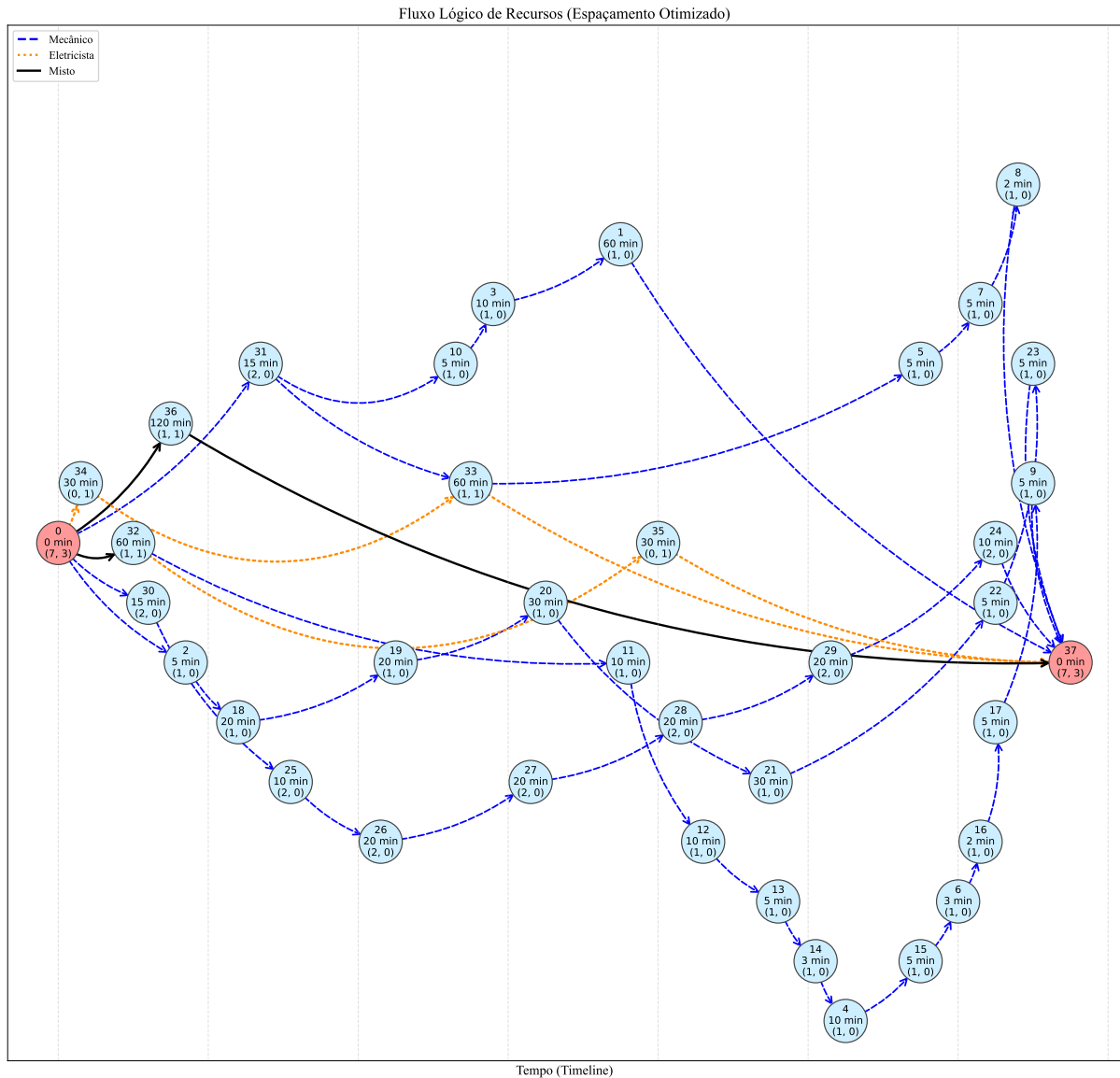
(z_{ij}), tratados na restrição 17, que foi adaptada no capítulo 3. Tal resultado foi obtido em um tempo de execução de 3,96 segundos, valor que representa 0,22% do tempo limite a disposição para resolução do problema (30 minutos). Se faz importante ressaltar também que o uso de memória foi estimado por uma biblioteca externa. Contudo, o projeto alcançou uma alocação de memória suficientemente baixa para que possa ser uma alternativa de implementação viável. Além disso, o *solver* explorou 2363 ramificações, evidenciando a complexidade combinatória do problema NP-Difícil, conforme discutido por Artigues, Demassey e Néron (2008) e Garey e Johnson (1975).

Esses dados corroboram a escolha das ferramentas de código aberto baseadas em Python, que, além de acessíveis, apresentaram robustez suficiente para lidar com a instância real da indústria automotiva estudada.

4.3 Análise do Cronograma Otimizado

O objetivo central do trabalho, definido na Equação (1) do modelo matemático, foi a minimização do tempo total de execução do projeto (*makespan*) de 38 atividades (reais e fictícias), cujo fluxo de recursos foi determinado pelo modelo e pode ser observado na Figura 4.

Figura 4 – Grafo de fluxo de recursos entre as atividades.

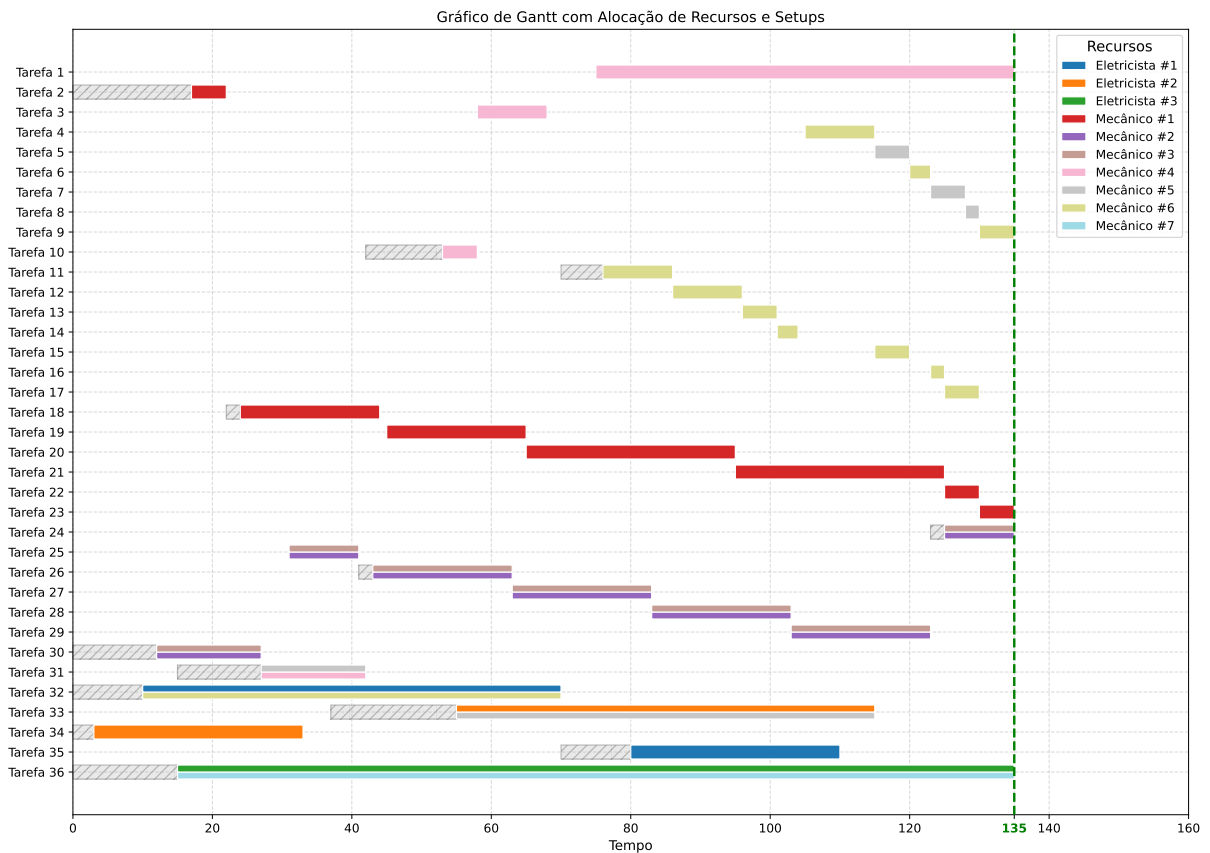


Fonte: O autor (2026).

No grafo acima, criado a partir dos valores da variável f_{ijk} , é possível observar o fluxo de recurso entre as atividades. Cada atividade é representada por um nó (circulo azul para atividades reais e vermelho para atividades fictícias). Nesses nós, estão representados o número da atividade, seu tempo de execução e a quantidade de recursos entre parênteses, de forma que o primeiro item dentro dos parênteses representa a quantidade de mecânicos e o segundo a quantidade de eletricitas necessários.

Após obtenção do grafo da Figura 4, o próximo passo foi visualizar a ordem cronológica ideal de execução obtida pelo modelo. A Figura 5 apresenta o Gráfico de Gantt gerado pela aplicação, ilustrando a distribuição temporal das atividades.

Figura 5 – Gráfico de Gantt do sequenciamento otimizado com tempos de setup.



Fonte: O autor (2026).

A análise da Figura 5 permite visualizar a influência dos tempos de deslocamento, representados pelas barras hachuradas em cinza. Evidencia-se que o modelo buscou sequenciar as atividades de forma a minimizar esses intervalos improdutivos, agrupando, sempre que as restrições permitiram, tarefas geograficamente próximas para um mesmo recurso.

Adicionalmente, o algoritmo de pós-processamento (em Python) permitiu a visualização individualizada dos recursos, explicitada na legenda. Porém, se faz importante ressaltar, que, como os as unidades de um mesmo recurso não são tratados de forma individualizada no modelo matemático, a atribuição específica apresentada é apenas uma das possibilidades viáveis de alocação individual, não sendo necessariamente única para atingir o *makespan* ótimo.

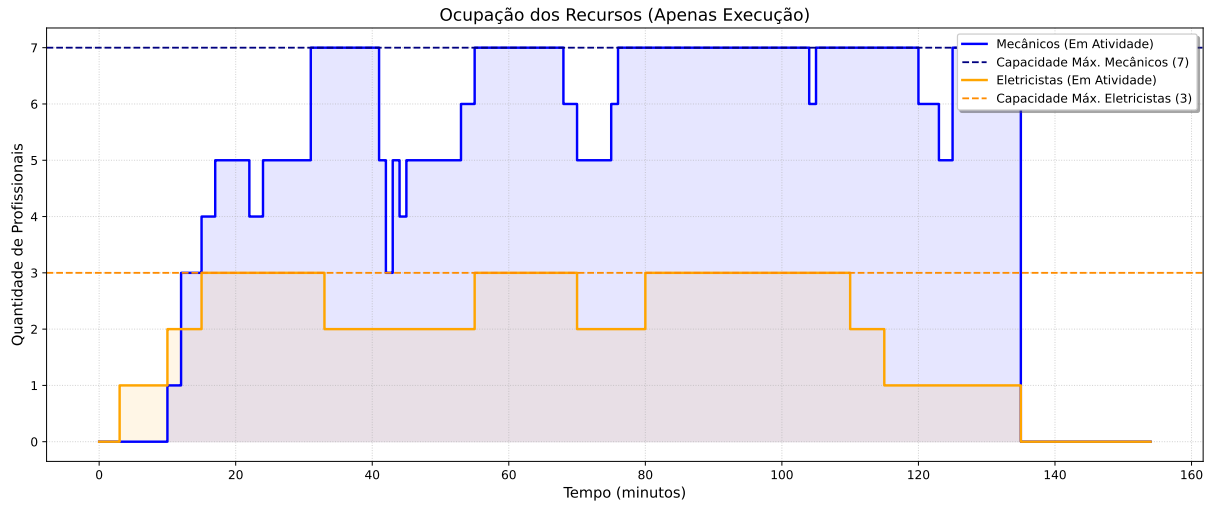
4.4 Alocação de Recursos

A disponibilidade de recursos, restrita a 7 mecânicos e 3 eletricitas, foi um fator determinante para o sequenciamento. O modelo respeitou corretamente as restrições de capacidade (Equações (7), (8) e (9)).

A análise detalhada dos dados de saída, através do *script* de pós processamento permite

observar a taxa de ocupação dos diferentes tipos de mão de obra, através da Figura 6:

Figura 6 – Perfil de ocupação da mão de obra (apenas tempos de execução).



Fonte: O autor (2026).

A Figura 6 ilustra o perfil de alocação de mão de obra ao longo do horizonte de planejamento. Se faz necessário pontuar que os níveis de ocupação apresentados referem-se apenas aos momentos de execução das atividades. Devido à formulação matemática adotada, os tempos de *setup* (z_{ij}) atuam como restrições de tempo de precedência, impondo um intervalo obrigatório entre tarefas, e não como consumo de recursos na variável de fluxo de recursos. Consequentemente, durante os deslocamentos, o gráfico não contabiliza a ocupação desta mão de obra, refletindo a lógica de otimização do modelo que trata esse período como tempo não produtivo. Entretanto, é notável que ainda sim os recursos obtiveram uma alta taxa de ocupação durante grande parte do tempo, como pode ser observado também na tabela 6:

Tabela 6 – Taxa de ocupação dos recursos.

Indicador	Mecânicos	Eletricistas
Capacidade Disponível (R_k)	7	3
Taxa produtiva (%)	80,42	74,07
Taxa de ocupação total(%)	97,67	94,81

*O *Makespan* (tempo total) considerado para o cálculo foi de 135 minutos.

Fonte: O autor (2026).

A partir da Tabela 6, nota-se que a taxa de ocupação total (considera tempos de *setup*) dos recursos foi de 97,67% para os mecânicos e de 94,81% para os eletricitistas, valores que demonstram baixa ociosidade de mão de obra. Essa métrica comprova a capacidade do modelo em maximizar o uso da equipe disponível e minimizar o tempo ocioso não planejado. Simultaneamente, índices

tão elevados alertam para a reduzida margem de folga do cronograma, indicando que o projeto pode estar vulnerável a riscos relacionados a atrasos e contratempos operacionais.

4.5 Cenário Alternativo

Com a finalidade de demonstrar a flexibilidade do modelo em adaptar-se a diferentes condições operacionais, elaborou-se um cenário alternativo mantendo as mesmas características e parâmetros do cenário principal, com exceção da quantidade de mecânicos, que foi incrementada para 8 colaboradores. O contingente de eletricitas não foi alterado, visto que o número de tarefas que exigem esse tipo de profissional já coincide com a capacidade disponível da equipe.

Este cenário foi proposto com o objetivo de diminuir a taxa de ocupação da mão de obra, visando mitigar a vulnerabilidade do projeto a eventuais atrasos.

Tabela 7 – Parâmetros de entrada do cenário alternativo.

Parâmetro	Quantidade
Linhas de produção	3
Operações	9
Tempo disponível	150 min
Atividades	36
Mecânicos	8
Eletricitas	3

Fonte: O autor (2026).

A Tabela 7 ratifica que os parâmetros fundamentais do cenário principal foram preservados, alterando-se exclusivamente a capacidade do recurso de mecânicos para 8 unidades.

4.5.1 Resultados do Novo Cenário

Para este novo cenário, o modelo matemático adaptado foi novamente executado através da biblioteca *Google OR-Tools*, seguindo os procedimentos metodológicos consolidados no Capítulo 3. Como resultado, obteve-se o mesmo *makespan* de 135 minutos do cenário base, conforme detalhado na Tabela 8:

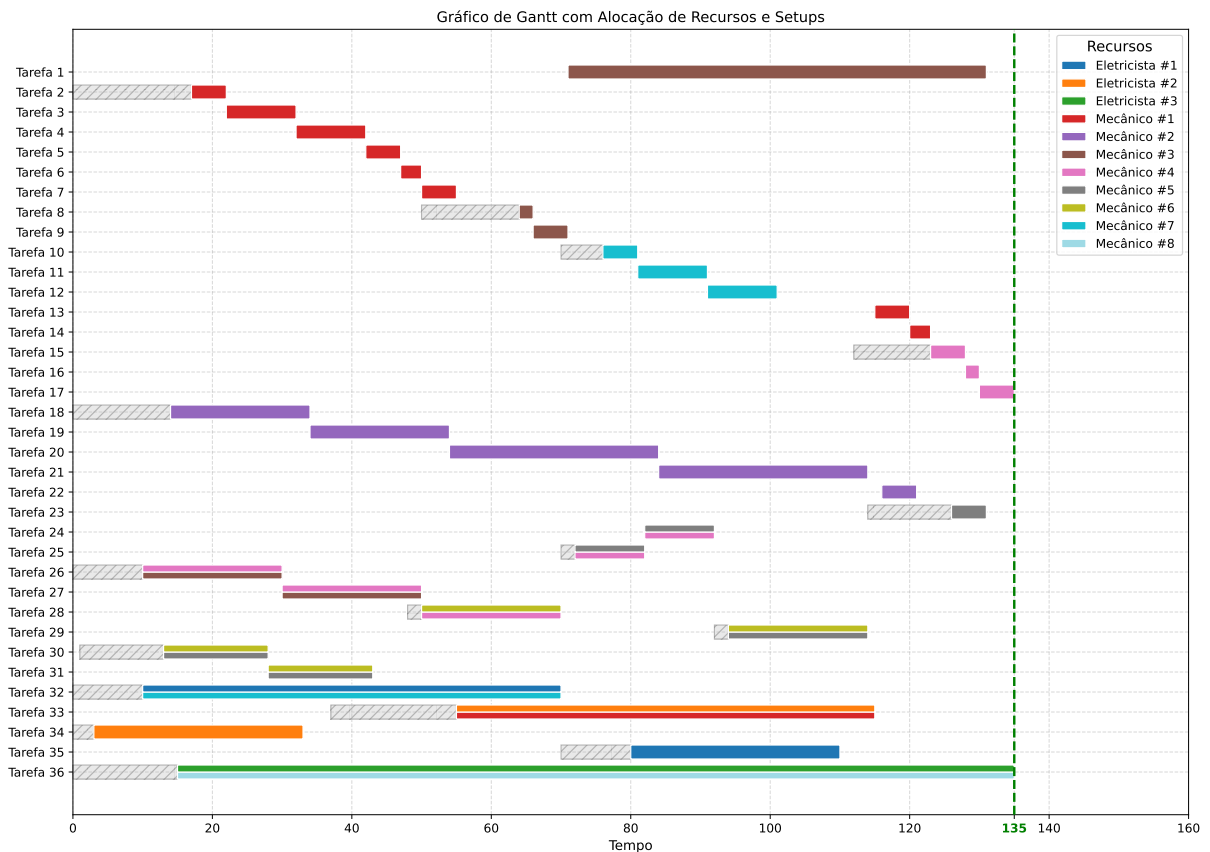
Tabela 8 – Resultado ótimo do cenário alternativo.

Indicador	Valor
Status da solução	Ótima
<i>makespan</i>	135 min
Tempo de Processamento (<i>Wall time</i>)	0,34 s
Uso de memória*	69,19 MB
ramificações	936

Fonte: O autor (2026).

A Tabela 8 demonstra que o modelo alcançou o tempo total de projeto de 135 minutos. Constata-se também que os indicadores de desempenho mantiveram-se eficientes, com baixo tempo de processamento e baixo consumo de memória (estimado pela biblioteca *psutil*), à semelhança dos resultados do cenário principal. A Figura 7 exibe o Gráfico de Gantt correspondente a este novo sequenciamento:

Figura 7 – Gráfico de Gantt do sequenciamento do cenário alternativo.

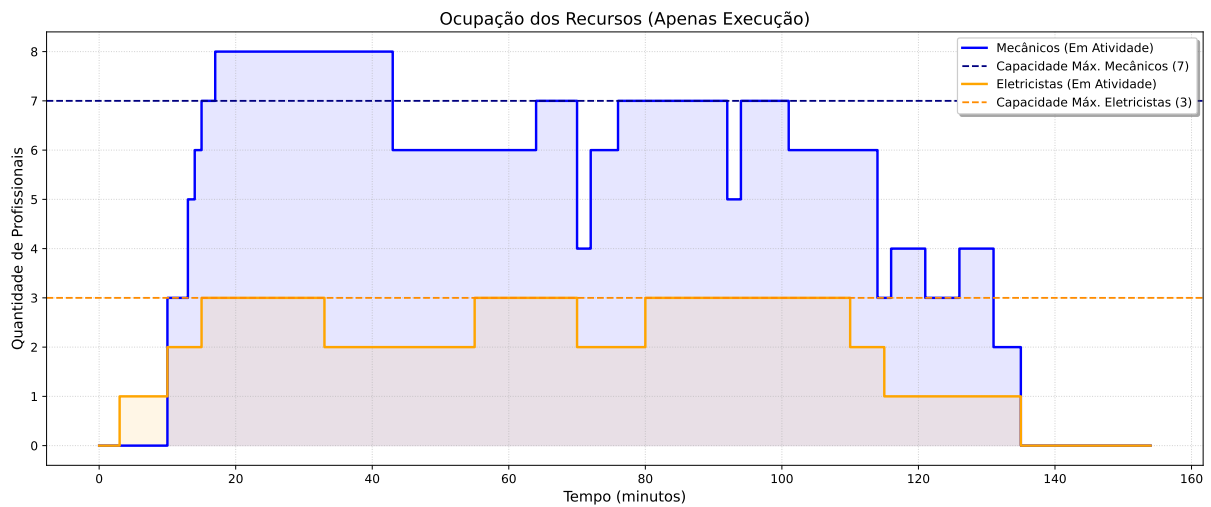


Fonte: O autor (2026).

A partir da Figura 7, nota-se que, embora o *makespan* tenha permanecido inalterado (limitado pela duração e tempo de *setup* da Atividade 36), as demais tarefas foram redistribuídas de forma a aproveitar o mecânico adicional escalado.

A referida redistribuição da força de trabalho pode ser visualmente confirmada pela análise do perfil de ocupação na Figura 8, apresentada a seguir:

Figura 8 – Perfil de ocupação da mão de obra para o cenário alternativo.



Fonte: O autor (2026).

Por fim, as alterações referentes à taxa de utilização da mão de obra foram quantificadas por meio dos algoritmos de pós-processamento construídos em *Python*, empregados para avaliar as variáveis e parâmetros do modelo. O balanço desta análise encontra-se na Tabela 9:

Tabela 9 – Taxa de ocupação dos recursos no cenário alternativo.

Indicador	Mecânicos	Eletricistas
Capacidade Disponível (R_k)	8	3
Taxa produtiva (%)	70,37	74,07
Taxa de ocupação total(%)	87,87	94,81

*O *Makespan* (tempo total) considerado para o cálculo foi de 135 minutos.

Fonte: O autor (2026).

A Tabela 9 atesta que a inserção de um mecânico adicional reduziu a taxa produtiva desta modalidade da equipe em aproximadamente 10,05%, em relação ao cenário anterior. Essa mitigação na sobrecarga dos recursos humanos confere maior resiliência ao cronograma, reduzindo os riscos associados à vulnerabilidade a atrasos operacionais.

5 Considerações Finais

O presente trabalho dedicou-se a abordar o Problema de Sequenciamento de Projeto com Restrições de Recursos (PSPRR) aplicado ao contexto da manutenção industrial de uma grande empresa do setor automobilístico. O objetivo central foi propor e validar um modelo de otimização matemática capaz de minimizar o tempo total de execução (*makespan*) de atividades de manutenção cíclicas, validando o uso de tecnologias de código aberto.

5.1 Conclusão

A implementação computacional do modelo matemático, adaptado da proposta de [Vieira e Rola \(2024\)](#) para incluir tempos de *setup* dependentes da sequência, demonstrou-se eficaz para o cenário estudado.

Esse resultado evidencia a importância da aplicação de ferramentas da Pesquisa Operacional no chão de fábrica, visto que, na indústria, em especial na automotiva, a eficiência e continuidade dos processos produtivos são fatores críticos para a competitividade das empresas, conforme explicitado na seção 2.1.

Nesse contexto, a utilização de técnicas como a PLIM apresenta vantagens que vão muito além daquelas observadas na teoria acadêmica. Essas ferramentas representam instrumentos práticos e decisivos capazes de transformar restrições complexas do chão de fábrica em diferenciais competitivos, resultando em operações mais enxutas e confiáveis.

A análise dos resultados permite estabelecer as seguintes conclusões principais:

5.1.1 Desempenho Computacional e Viabilidade de *Softwares* de Código Aberto

Na seção 4.2 é apresentado que o tempo de processamento para encontrar a solução ótima foi de apenas 3,96 segundos, o que representou aproximadamente 0,22% do tempo limite configurado para a execução da simulação. Aliado a um consumo aproximado de memória aceitável (cerca de 58 MB), **conclui-se que as tecnologias de código aberto baseadas em Python e Google OR-Tools são robustas e plenamente capazes de substituir ou complementar soluções comerciais de alto custo na indústria**, seja na execução do modelo, processamento e tratamentos dos dados ou na obtenção recursos visuais como gráficos e grafos, a fim de auxiliar os tomadores de decisão de forma estratégica.

5.1.2 Eficiência Operacional

A solução ótima encontrada resultou em um *makespan* de 135 minutos. Este valor é inferior ao limite de 150 minutos estipulado pela equipe de planejamento, garantindo que todas as atividades de manutenção sejam concluídas dentro da janela de tempo de parada da linha de produção, evitando prejuízos operacionais, como apresentado na seção 4.2.

5.1.3 Otimização de Recursos e Análise de Cenários

No cenário principal, o modelo promoveu uma alta taxa de ocupação total (incluindo tempos de deslocamento) dos recursos, superior a 94% tanto para mecânicos quanto para eletricitas, como pode ser observado na Tabela 6 da Seção 4.4. Isso indica que foi encontrada uma solução com baixa ociosidade e alto aproveitamento da mão de obra disponível, validando a capacidade do algoritmo em lidar com as restrições de precedência e disponibilidade simultaneamente.

Adicionalmente, a simulação do cenário alternativo (Seção 4.5) evidenciou a flexibilidade e o potencial analítico da ferramenta. Ao incrementar a capacidade de mão de obra mecânica em uma unidade e manter o contingente de eletricitas inalterado, o modelo foi capaz de redistribuir as tarefas de forma eficiente. Embora o tempo total de execução (*makespan*) tenha se mantido no patamar ótimo de 135 minutos, observou-se uma redução considerável na taxa de ocupação produtiva da equipe de mecânicos. Esse resultado atesta a utilidade do modelo não apenas para o sequenciamento, mas também como uma ferramenta de apoio à decisão para o dimensionamento tático de equipes, permitindo aos gestores mitigar a vulnerabilidade do sistema a eventuais atrasos.

Em suma, a pesquisa confirmou que **a aplicação de Pesquisa Operacional, suportada por ferramentas de código aberto, oferece um meio confiável e eficiente para orientar a tomada de decisão da equipe de manutenção em problemas do tipo PSPRR**, superando as limitações do planejamento manual.

5.2 Trabalhos Futuros

Apesar dos resultados satisfatórios, o modelo apresenta oportunidades de aprimoramento. Como sugestão para a continuidade desta pesquisa e trabalhos futuros, propõe-se incorporar abordagens estocásticas para considerar a variabilidade nos tempos de execução das atividades e nos tempos de deslocamento, tornando o cronograma mais resiliente a imprevistos comuns no chão de fábrica.

Ademais, propõe-se também a implementação de algoritmos heurísticos, haja visto que, este projeto envolveu um problema de 36 atividades reais, e, segundo [Koné et al. \(2011\)](#), o limite sugerido de atividades envolvidas em problemas de PLIM é de cerca de 30 atividades.

Por fim, é razoável que em trabalhos futuros haja a reintegração das restrições financeiras do modelo original, para se realizar análises entre o custo de contratação de mão de obra extra e

a redução do tempo de parada da linha, por exemplo.

Referências

ANAND, R.; AGGARWAL, D.; CHAHAR, V. A comparative analysis of optimization solvers. *Journal of Information & Optimization Sciences*, 07 2017. 5, 12

ARTIGUES, C.; DEMASSEY, S.; NÉRON, E. *Resource-Constrained Project Scheduling: Models, Algorithms, Extensions and Applications*. John Wiley & Sons, Ltd, 2008. 21-85 p. ISBN 9780470611227. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470611227.fmatter>>. 7, 10, 22

ARTIGUES, C.; MICHELON, P.; REUSSER, S. Insertion techniques for static and dynamic resource-constrained project scheduling. *European Journal of Operational Research*, v. 149, n. 2, p. 249–267, 2003. ISSN 0377-2217. Sequencing and Scheduling. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0377221702007580>>. 7, 11

BLAZEWICZ, J.; LENSTRA, J.; KAN, A. Scheduling subject to resource constraints: classification and complexity. *Discrete Applied Mathematics*, v. 5, n. 1, p. 11–24, 1983. ISSN 0166-218X. Disponível em: <<https://www.sciencedirect.com/science/article/pii/0166218X83900124>>. 7

BRUCKER, P. et al. A branch and bound algorithm for the resource-constrained project scheduling problem. *European Journal of Operational Research*, v. 107, n. 2, p. 272–288, 1998. ISSN 0377-2217. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0377221797003354>>. 11

CARRUTHERS, J. A.; BATTERSBY, A. Advances in critical path methods. *OR*, Operational Research Society, v. 17, n. 4, p. 359–380, 1966. ISSN 14732858. Disponível em: <<http://www.jstor.org/stable/3007441>>. 11

CHEN, D.-S.; BATSON, R. G.; DANG, Y. *Applied Integer Programming: Modeling and Solution*. John Wiley & Sons, Ltd, 2009. 1-179 p. ISBN 9781118166000. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118166000.ch1>>. 6

CHRISTOFIDES, N. Worst-case analysis of a new heuristic for the traveling salesman problem. *Carnegie Mellon University*, v. 3, p. 10, 03 2022. 10

CHRISTOFIDES, N.; ALVAREZ-VALDES, R.; TAMARIT, J. M. Project scheduling with resource constraints: A branch and bound approach. *European Journal of Operational Research*, v. 29, n. 3, p. 262–273, 1987. Disponível em: <<https://EconPapers.repec.org/RePEc:eee:ejores:v:29:y:1987:i:3:p:262-273>>. 11

DANTZIG, G. *Linear Programming and Extensions*. Princeton University Press, 1963. (Princeton landmarks in mathematics and physics). ISBN 9780691080000. Disponível em: <https://books.google.com.br/books?id=_wvAAAAMAAJ>. 6

DEMEULEMEESTER, E.; HERROELEN, W. A branch-and-bound procedure for the multiple resource-constrained project scheduling problem. *Katholieke Universiteit Leuven, Open Access publications from Katholieke Universiteit Leuven*, v. 38, 12 1992. 11

DREWES, S.; ULBRICH, S. *Mixed integer second order cone programming*. [S.l.]: Verlag Dr. Hut Germany, 2009. 6

DUAN, S. et al. The applications of hybrid approach combining exact method and evolutionary algorithm in combinatorial optimization. *Journal of Computational Design and Engineering*, v. 10, n. 3, p. 934–946, 04 2023. ISSN 2288-5048. Disponível em: <<https://doi.org/10.1093/jcde/qwad029>>. 11, 12

DUFFUAA, S.; RAOUF, A.; CAMPBELL, J. *Planning and Control of Maintenance Systems: Modeling and Analysis*. Wiley, 1999. ISBN 9780471179818. Disponível em: <https://books.google.com.br/books?id=Yg4_fyoa0WkC>. 1, 4

FIACCO, A. V. *Encyclopedia of Operations Research and Management Science*. Boston, MA: Springer US, 2013. 1053-1062 p. ISBN 978-1-4419-1153-7. Disponível em: <https://doi.org/10.1007/978-1-4419-1153-7_682>. 6

FOULDS, L. R. The heuristic problem-solving approach. *Journal of the Operational Research Society*, v. 34, p. 927–934, 1983. 10

GAREY, M. R.; JOHNSON, D. S. Complexity results for multiprocessor scheduling under resource constraints. *SIAM Journal on Computing*, v. 4, n. 4, p. 397–411, 1975. Disponível em: <<https://doi.org/10.1137/0204035>>. 7, 22

GEARHART, J. L. et al. *Comparison of open-source linear programming solvers*. [S.l.], 2013. Disponível em: <<https://www.osti.gov/biblio/1104761>>. 12

GOLDBARG, M.; LUNA, H. *Otimização combinatória e programação linear: modelos e algoritmos*. Elsevier, 2005. ISBN 9788535215205. Disponível em: <<https://books.google.com.br/books?id=Q-bNGAAACAAJ>>. 4, 5

HAO, J.; HO, T. K. Machine learning made easy: A review of scikit-learn package in python programming language. *Journal of Educational and Behavioral Statistics*, v. 44, n. 3, p. 348–361, 2019. Disponível em: <<https://doi.org/10.3102/1076998619832248>>. 12

HARTMANN, S. Project scheduling with resource capacities requests varying with time. *Flexible Services Manufacturing Journal*, p. 2–5, 2013. 7

HIPPEL, E. von. *Democratizing Innovation*. [S.l.]: MIT Press, 2006. v. 48. ISBN 9780262285636. 11

HOLWEG, M. The evolution of competition in the automotive industry. In: *Build to order: The road to the 5-day car*. [S.l.]: Springer, 2008. p. 13–34. 1, 3

IOMMAZZO, G. et al. Learning to configure mathematical programming solvers by mathematical programming. In: _____. *Learning and Intelligent Optimization*. Springer International Publishing, 2020. p. 377–389. ISBN 9783030535520. Disponível em: <http://dx.doi.org/10.1007/978-3-030-53552-0_34>. 12

KARP, R. M. *The Probabilistic Analysis of some Combinational Search Algorithms*. [S.l.], 1976. Disponível em: <<http://www2.eecs.berkeley.edu/Pubs/TechRpts/1976/28848.html>>. 10

KOLISCH, R.; PADMAN, R. An integrated survey of deterministic project scheduling. *Omega*, v. 29, n. 3, p. 249–272, 2001. ISSN 0305-0483. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0305048300000463>>. 1, 7

KONÉ, O. et al. Event-based milp models for resource-constrained project scheduling problems. *Computers & Operations Research*, v. 38, n. 1, p. 3–13, 2011. ISSN 0305-0548. Project Management and Scheduling. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0305054809003360>>. 7, 11, 16, 30

KROLAK, P.; FELTS, W.; MARBLE, G. A man-machine approach toward solving the traveling salesman problem. *Commun. ACM*, Association for Computing Machinery, New York, NY, USA, v. 14, n. 5, p. 327–334, May 1971. ISSN 0001-0782. Disponível em: <<https://doi.org/10.1145/362588.362593>>. 10

LAZIMY, R. Mixed-integer quadratic programming. *Mathematical Programming*, v. 22, p. 332–349, 1982. 6

LIN, S.; KERNIGHAN, B. W. An effective heuristic algorithm for the traveling-salesman problem. *Operations Research*, v. 21, n. 2, p. 498–516, 1973. ISSN 0030364X, 15265463. Disponível em: <<http://www.jstor.org/stable/169020>>. 10

MOUBRAY, J. *Reliability-centered Maintenance*. Industrial Press, 2001. (G - Reference, Information and Interdisciplinary Subjects Series). ISBN 9780831131463. Disponível em: <<https://books.google.com.br/books?id=bNCVF0B7vpIC>>. 3

MUNISAMY, R. *Essential Python Libraries for Data Analysis and Machine Learning*. 2025. Disponível em: <<https://medium.com/@ramarajmsu/essential-python-libraries-for-data-analysis-and-machine-learning-1f47c12fceef>>. 12

NAKAJIMA, S. *TPM Development Program: Implementing Total Productive Maintenance*. Productivity Press, 1989. (Preventative Maintenance Series). ISBN 9780915299379. Disponível em: <<https://books.google.com.br/books?id=Q6F9QgAACAAJ>>. 3

OSMAN, I.; LAPORTE, G. Metaheuristics: A bibliography. *Annals of Operational Research*, v. 63, p. 513–628, 10 1996. 10

PENHA, R. et al. Produção científica sobre resource-constrained project scheduling problem: Um estudo bibliométrico e bibliográfico. *Revista de Gestão e Projetos*, v. 8, n. 2, p. 71–86, 2017. Disponível em: <<https://periodicos.uninove.br/gep/article/view/9668>>. 11

PERRON, L.; DIDIER, F.; GAY, S. The CP-SAT-LP Solver. In: YAP, R. H. C. (Ed.). *29th International Conference on Principles and Practice of Constraint Programming (CP 2023)*. Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. (Leibniz International Proceedings in Informatics (LIPIcs), v. 280), p. 3:1–3:2. ISBN 978-3-95977-300-3. ISSN 1868-8969. Disponível em: <<https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.CP.2023.3>>. 12

PRITSKER, A. A. B.; WAITERS, L. J.; WOLFE, P. M. Multiproject scheduling with limited resources: A zero-one programming approach. *Management Science*, v. 16, n. 1, p. 93–108, 1969. 11

RAYMOND, E. *The Cathedral & the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. O'Reilly Media, 2001. (O'Reilly Series). ISBN 9780596001087. Disponível em: <<https://books.google.com.br/books?id=W2t2d2KP6HsC>>. 1, 11

- ROSSI, F.; van Beek, P.; WALSH, T. Chapter 4 constraint programming. In: van Harmelen, F.; LIFSCHITZ, V.; PORTER, B. (Ed.). *Handbook of Knowledge Representation*. Elsevier, 2008, (Foundations of Artificial Intelligence, v. 3). p. 181–211. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1574652607030040>. 6
- SANTOS, H. G.; TOFFOLO, T. Mixed integer linear programming with python. *COINOR Computational Infrastructure for Operations Research*, 2020. 12
- SOUZA, C. da Silva Vieira Camêlo de; WELERSON, J. V. A. C. M. O problema de sequenciamento em projeto com restrição de recurso: um estudo bibliométrico. *REVISTA DELOS*, v. 18, 01 2025. 11
- TALBOT, F. B.; PATTERSON, J. H. An Efficient Integer Programming Algorithm with Network Cuts for Solving Resource-Constrained Scheduling Problems. *Management Science*, v. 24, n. 11, p. 1163–1174, July 1978. Disponível em: <https://ideas.repec.org/a/inm/ormnsc/v24y1978i11p1163-1174.html>. 11
- TEAM, A. *Python for data science: A complete guide*. 2025. Disponível em: <https://www.anaconda.com/guides/python-for-data-science>. 12
- TÉCNICAS, A. B. de N. *NBR 5462: Confiabilidade e manutenibilidade*. Rio de Janeiro, 1994. 37 p. 3
- VIEIRA, C. da S.; ROLA, F. S. C. Proposições de modelos matemáticos para o problema de sequenciamento em projetos com restrição de recursos e análise de desempenho. *REVISTA FOCO*, v. 17, n. 6, p. 01–17, jun. 2024. Disponível em: <https://ojs.focopublicacoes.com.br/foco/article/view/5350>. vii, viii, 2, 7, 11, 15, 18, 21, 29

Apêndices

APÊNDICE A – Matriz de Setup (parte 1)

Tabela 10 – Matriz de Setup (Parte 1: Destinos 0 a 18)

Orig.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
0	0,00	17,17	17,17	17,17	17,17	17,17	17,17	17,17	17,17	17,17	17,17	17,17	17,17	17,17	17,17	17,17	17,17	17,17	17,17
1	17,17	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
2	17,17	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
3	17,17	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
4	17,17	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
5	17,17	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
6	17,17	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
7	17,17	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
8	17,17	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
9	17,17	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
10	17,17	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
11	17,17	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
12	17,17	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
13	17,17	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
14	17,17	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
15	17,17	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
16	17,17	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
17	17,17	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
18	14,67	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50
19	14,67	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50
20	14,67	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50
21	14,67	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50

Continua na próxima página...

Tabela 10 – continuação da página anterior

Orig.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
22	14,67	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50
23	14,67	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50	2,50
24	7,50	11,50	11,50	11,50	11,50	11,50	11,50	11,50	11,50	11,50	11,50	11,50	11,50	11,50	11,50	11,50	11,50	11,50	11,50
25	7,50	11,50	11,50	11,50	11,50	11,50	11,50	11,50	11,50	11,50	11,50	11,50	11,50	11,50	11,50	11,50	11,50	11,50	11,50
26	10,42	14,42	14,42	14,42	14,42	14,42	14,42	14,42	14,42	14,42	14,42	14,42	14,42	14,42	14,42	14,42	14,42	14,42	14,42
27	10,42	14,42	14,42	14,42	14,42	14,42	14,42	14,42	14,42	14,42	14,42	14,42	14,42	14,42	14,42	14,42	14,42	14,42	14,42
28	10,42	14,42	14,42	14,42	14,42	14,42	14,42	14,42	14,42	14,42	14,42	14,42	14,42	14,42	14,42	14,42	14,42	14,42	14,42
29	10,42	14,42	14,42	14,42	14,42	14,42	14,42	14,42	14,42	14,42	14,42	14,42	14,42	14,42	14,42	14,42	14,42	14,42	14,42
30	12,50	11,50	11,50	11,50	11,50	11,50	11,50	11,50	11,50	11,50	11,50	11,50	11,50	11,50	11,50	11,50	11,50	11,50	11,50
31	12,50	11,50	11,50	11,50	11,50	11,50	11,50	11,50	11,50	11,50	11,50	11,50	11,50	11,50	11,50	11,50	11,50	11,50	11,50
32	10,50	6,67	6,67	6,67	6,67	6,67	6,67	6,67	6,67	6,67	6,67	6,67	6,67	6,67	6,67	6,67	6,67	6,67	6,67
33	18,00	0,83	0,83	0,83	0,83	0,83	0,83	0,83	0,83	0,83	0,83	0,83	0,83	0,83	0,83	0,83	0,83	0,83	0,83
34	3,67	15,83	15,83	15,83	15,83	15,83	15,83	15,83	15,83	15,83	15,83	15,83	15,83	15,83	15,83	15,83	15,83	15,83	15,83
35	4,50	16,67	16,67	16,67	16,67	16,67	16,67	16,67	16,67	16,67	16,67	16,67	16,67	16,67	16,67	16,67	16,67	16,67	16,67
36	15,50	1,67	1,67	1,67	1,67	1,67	1,67	1,67	1,67	1,67	1,67	1,67	1,67	1,67	1,67	1,67	1,67	1,67	1,67
37	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00

APÊNDICE B – Matriz de Setup (parte 2)

Tabela 11 – Matriz de Setup (Parte 2: Destinos 19 a 37)

Orig.	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37
0	14,67	14,67	14,67	14,67	14,67	7,50	7,50	10,42	10,42	10,42	10,42	12,50	12,50	10,50	18,00	3,67	4,50	15,50	0,00
1	0,00	2,50	2,50	2,50	2,50	2,50	2,50	11,50	11,50	14,42	14,42	14,42	14,42	11,50	11,50	6,67	0,83	15,83	16,67
2	0,00	2,50	2,50	2,50	2,50	2,50	2,50	11,50	11,50	14,42	14,42	14,42	14,42	11,50	11,50	6,67	0,83	15,83	16,67
3	0,00	2,50	2,50	2,50	2,50	2,50	2,50	11,50	11,50	14,42	14,42	14,42	14,42	11,50	11,50	6,67	0,83	15,83	16,67
4	0,00	2,50	2,50	2,50	2,50	2,50	2,50	11,50	11,50	14,42	14,42	14,42	14,42	11,50	11,50	6,67	0,83	15,83	16,67
5	0,00	2,50	2,50	2,50	2,50	2,50	2,50	11,50	11,50	14,42	14,42	14,42	14,42	11,50	11,50	6,67	0,83	15,83	16,67
6	0,00	2,50	2,50	2,50	2,50	2,50	2,50	11,50	11,50	14,42	14,42	14,42	14,42	11,50	11,50	6,67	0,83	15,83	16,67
7	0,00	2,50	2,50	2,50	2,50	2,50	2,50	11,50	11,50	14,42	14,42	14,42	14,42	11,50	11,50	6,67	0,83	15,83	16,67
8	0,00	2,50	2,50	2,50	2,50	2,50	2,50	11,50	11,50	14,42	14,42	14,42	14,42	11,50	11,50	6,67	0,83	15,83	16,67
9	0,00	2,50	2,50	2,50	2,50	2,50	2,50	11,50	11,50	14,42	14,42	14,42	14,42	11,50	11,50	6,67	0,83	15,83	16,67
10	0,00	2,50	2,50	2,50	2,50	2,50	2,50	11,50	11,50	14,42	14,42	14,42	14,42	11,50	11,50	6,67	0,83	15,83	16,67
11	0,00	2,50	2,50	2,50	2,50	2,50	2,50	11,50	11,50	14,42	14,42	14,42	14,42	11,50	11,50	6,67	0,83	15,83	16,67
12	0,00	2,50	2,50	2,50	2,50	2,50	2,50	11,50	11,50	14,42	14,42	14,42	14,42	11,50	11,50	6,67	0,83	15,83	16,67
13	0,00	2,50	2,50	2,50	2,50	2,50	2,50	11,50	11,50	14,42	14,42	14,42	14,42	11,50	11,50	6,67	0,83	15,83	16,67
14	0,00	2,50	2,50	2,50	2,50	2,50	2,50	11,50	11,50	14,42	14,42	14,42	14,42	11,50	11,50	6,67	0,83	15,83	16,67
15	0,00	2,50	2,50	2,50	2,50	2,50	2,50	11,50	11,50	14,42	14,42	14,42	14,42	11,50	11,50	6,67	0,83	15,83	16,67
16	0,00	2,50	2,50	2,50	2,50	2,50	2,50	11,50	11,50	14,42	14,42	14,42	14,42	11,50	11,50	6,67	0,83	15,83	16,67
17	0,00	2,50	2,50	2,50	2,50	2,50	2,50	11,50	11,50	14,42	14,42	14,42	14,42	11,50	11,50	6,67	0,83	15,83	16,67
18	0,00	0,00	0,00	0,00	0,00	9,50	9,50	12,42	12,42	12,42	12,42	9,50	9,50	4,17	3,33	13,33	14,17	0,83	0,00
19	0,00	0,00	0,00	0,00	0,00	9,50	9,50	12,42	12,42	12,42	12,42	9,50	9,50	4,17	3,33	13,33	14,17	0,83	0,00
20	0,00	0,00	0,00	0,00	0,00	9,50	9,50	12,42	12,42	12,42	12,42	9,50	9,50	4,17	3,33	13,33	14,17	0,83	0,00
21	0,00	0,00	0,00	0,00	0,00	9,50	9,50	12,42	12,42	12,42	12,42	9,50	9,50	4,17	3,33	13,33	14,17	0,83	0,00

Continua na próxima página...

Tabela 11 – continuação da página anterior

Orig.	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37
22	0,00	0,00	0,00	0,00	0,00	9,50	9,50	12,42	12,42	12,42	12,42	9,50	9,50	4,17	3,33	13,33	14,17	0,83	0,00
23	0,00	0,00	0,00	0,00	0,00	9,50	9,50	12,42	12,42	12,42	12,42	9,50	9,50	4,17	3,33	13,33	14,17	0,83	0,00
24	9,50	9,50	9,50	9,50	9,50	0,00	0,00	2,92	2,92	2,92	2,92	0,00	0,00	7,50	12,00	6,17	7,00	10,50	0,00
25	9,50	9,50	9,50	9,50	9,50	0,00	0,00	2,92	2,92	2,92	2,92	0,00	0,00	7,50	12,00	6,17	7,00	10,50	0,00
26	12,42	12,42	12,42	12,42	12,42	2,92	2,92	0,00	0,00	0,00	0,00	2,92	2,92	15,42	14,92	9,08	9,92	13,42	0,00
27	12,42	12,42	12,42	12,42	12,42	2,92	2,92	0,00	0,00	0,00	0,00	2,92	2,92	15,42	14,92	9,08	9,92	13,42	0,00
28	12,42	12,42	12,42	12,42	12,42	2,92	2,92	0,00	0,00	0,00	0,00	2,92	2,92	15,42	14,92	9,08	9,92	13,42	0,00
29	12,42	12,42	12,42	12,42	12,42	2,92	2,92	0,00	0,00	0,00	0,00	2,92	2,92	15,42	14,92	9,08	9,92	13,42	0,00
30	9,50	9,50	9,50	9,50	9,50	0,00	0,00	2,92	2,92	2,92	2,92	0,00	0,00	7,50	12,00	6,17	7,00	10,50	0,00
31	9,50	9,50	9,50	9,50	9,50	0,00	0,00	2,92	2,92	2,92	2,92	0,00	0,00	7,50	12,00	6,17	7,00	10,50	0,00
32	4,17	4,17	4,17	4,17	4,17	7,50	7,50	15,42	15,42	15,42	15,42	7,50	7,50	0,00	12,50	7,67	10,00	5,00	0,00
33	3,33	3,33	3,33	3,33	3,33	12,00	12,00	14,92	14,92	14,92	14,92	12,00	12,00	12,50	0,00	16,67	17,50	2,50	0,00
34	13,33	13,33	13,33	13,33	13,33	6,17	6,17	9,08	9,08	9,08	9,08	6,17	6,17	7,67	16,67	0,00	0,83	14,17	0,00
35	14,17	14,17	14,17	14,17	14,17	7,00	7,00	9,92	9,92	9,92	9,92	7,00	7,00	10,00	17,50	0,83	0,00	15,00	0,00
36	0,83	0,83	0,83	0,83	0,83	10,50	10,50	13,42	13,42	13,42	13,42	10,50	10,50	5,00	2,50	14,17	15,00	0,00	0,00
37	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00