



**FUNDAÇÃO EDSON QUEIROZ
UNIVERSIDADE DE FORTALEZA – UNIFOR
VICE-REITORIA DE ENSINO DE GRADUAÇÃO E PÓS-
GRADUAÇÃO – VRE
DIRETORIA DE PÓS-GRADUAÇÃO – DPG
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA
APLICADA – PPGIA**

**MODELO DE OTIMIZAÇÃO PARA O ESCALONAMENTO DE
MOTORISTAS NO TRANSPORTE RODOVIÁRIO SOB
REGULAMENTAÇÃO EUROPEIA**

FRANCISCO VANDO CARNEIRO MOREIRA

**FORTALEZA – CE
NOVEMBRO, 2025**

Francisco Vando Carneiro Moreira

**Modelo de Otimização para o Escalonamento de
Motoristas no Transporte Rodoviário sob Regulamentação
Europeia**

Exame de Qualificação apresentada ao Programa de Pós-Graduação em Informática Aplicada – PPGIA da Universidade de Fortaleza – UNIFOR, como parte dos requisitos para obtenção do título de Mestre em Informática Aplicada.

Universidade de Fortaleza – UNIFOR

Orientador: Prof. Dr. Plácido Rogério Pinheiro

Fortaleza – CE
Novembro, 2025

Ao meu pai.

Agradecimentos

Agradeço, primeiramente, ao Prof. Dr. Plácido Rogério Pinheiro, cuja orientação firme, dedicação acadêmica e confiança constante foram determinantes para a concretização desta pesquisa. Sua capacidade de incentivar, esclarecer e apontar caminhos, mesmo nos momentos mais desafiadores, foi essencial para que eu pudesse avançar com segurança e convicção. Sou profundamente grato por acreditar no meu potencial e por me acompanhar de forma tão generosa nesta jornada.

Expresso também minha sincera gratidão ao Prof. Dr. Valdísio Viana e ao Prof. Dr. Antonio Clécio Thomas, pelo apoio, pelas contribuições valiosas e pela disponibilidade em compartilhar conhecimento e experiência. Suas observações e estímulos fortaleceram-me e ampliaram meu olhar acadêmico em outros momentos da minha jornada de vida.

À minha esposa Carolina Araujo, deixo meu agradecimento mais especial. Pelo apoio incondicional, pela compreensão diante das ausências, pelo estímulo constante e pela serenidade com que dividiu os desafios desta caminhada. Nada disso seria possível sem a sua presença carinhosa, firme e essencial.

Aos meus pais, agradeço pelo amor, pelos valores e por tudo o que construíram em mim ao longo da vida. Em especial, dedico este momento à memória do meu pai, cuja força, exemplo e ensinamentos permanecem vivos em cada conquista. Sua ausência física jamais apagou sua presença na minha trajetória. Este trabalho também é dele.

A todos que, direta ou indiretamente, contribuíram para que este estudo se tornasse realidade, deixo meu mais profundo reconhecimento e gratidão.

“Tudo o que sou carrega um pouco de ti.”

- VM

Resumo

Este trabalho apresenta o desenvolvimento de um simulador avançado para o escalonamento de motoristas no transporte rodoviário europeu, fundamentado em um modelo de Programação Linear Inteira (PLI) que incorpora integralmente as exigências legais estabelecidas pelo Regulamento (CE) n.º 561/2006, pela Diretiva 2002/15/CE e pelo Regulamento (UE) n.º 165/2014. O modelo matemático foi formulado de modo a representar, de forma rigorosa, todas as restrições diárias, semanais e quinzenais de condução, pausas e períodos de descanso obrigatórios, incluindo dependências temporais complexas e janelas móveis de conformidade.

A implementação computacional foi realizada em Python, utilizando o solver CP-SAT do OR-Tools, e integrada a uma interface interativa desenvolvida em Streamlit, permitindo a configuração de cenários, seleção dinâmica de restrições, definição de funções objetivo e visualização detalhada dos resultados. Além do modo exato de resolução, o simulador incorpora abordagens heurísticas e uma estratégia matheurística baseada em *Large Neighborhood Search* (LNS), possibilitando a análise do comportamento do modelo em diferentes regimes de complexidade e escalabilidade.

Adicionalmente, o trabalho propõe a integração experimental de modelos de aprendizado de máquina supervisionado, utilizados para guiar decisões locais de alocação (f_1) e a seleção de vizinhanças no LNS (f_2), preservando sempre mecanismos de *fallback* heurístico para garantir robustez e estabilidade operacional. O simulador disponibiliza um conjunto abrangente de indicadores de desempenho (KPIs), incluindo cobertura global da demanda, eficiência operacional, risco operacional, estabilidade temporal e custo estimado, bem como visualizações analíticas e mapas de calor que facilitam a interpretação dos resultados.

Os experimentos computacionais conduzidos em horizontes de curto e médio prazo demonstram a capacidade do simulador em identificar soluções viáveis e analisar cenários inviáveis sob diferentes combinações de restrições, fornecendo suporte à tomada de decisão estratégica, tática e operacional. Como principal contribuição, este trabalho consolida um ambiente unificado de modelagem, simulação e análise para o escalonamento de motoristas sob a legislação europeia, avançando o estado da arte ao combinar otimização matemática, heurísticas, matheurísticas e aprendizado de máquina em um único framework experimental.

Palavras-chave: Programação Linear Inteira; Escalonamento de Motoristas; Regulamento (CE) n.º 561/2006; Otimização; OR-Tools; Matheurísticas; Simulação.

Abstract

This work presents the development of an Integer Linear Programming (ILP) model for driver scheduling in European road transport operations, fully aligned with the requirements established by Regulation (EC) No. 561/2006, Directive 2002/15/EC, and Regulation (EU) No. 165/2014. The model accurately represents all daily, weekly, and biweekly constraints regarding driving limits, mandatory breaks, and rest periods, incorporating complex temporal dependencies and rolling compliance windows. The computational implementation was developed in Python using the OR-Tools CP-SAT solver, together with an interactive Streamlit interface for scenario configuration, model execution, and operational indicator visualization. Experiments conducted over 7-, 15-, and 30-day planning horizons demonstrated high computational efficiency, consistent optimal solutions, and strong alignment between operational demand and allocated drivers. The results show that the proposed model is robust, scalable, and applicable to real-world road transport operations, providing support for strategic, tactical, and operational decision-making. Furthermore, the research advances the state of the art by offering a complete mathematical formulation for driver scheduling under European legislation, contributing to the academic literature and opening opportunities for future studies on integrated optimization, stochastic modeling, and SaaS-based decision-support systems for the transport sector.

Keywords: Integer Linear Programming; Driver Scheduling; European Regulation 561/2006; Optimization; OR-Tools; Logistics.

Lista de Abreviaturas e Siglas

ABNT Associação Brasileira de Normas Técnicas

CE Comunidade Europeia

CP-SAT Constraint Programming – Satisfiability Solver

EC European Community

EU European Union

FMS Fleet Management System

ILP Integer Linear Programming

LP Linear Programming

OR Operations Research

OR-Tools Optimization Tools (Google)

TMS Transport Management System

PLI Programação Linear Inteira

VRP Vehicle Routing Problem

List of Symbols

$x_{d,t}$ Motorista d trabalha no período t

$r_{d,t}$ Motorista d está em descanso no período t

$h_{d,t}$ Condução acumulada no período t

$s_{d,t}$ Início de jornada no período t

$z_{d,w}$ Uso de extensão diária na semana w

Δ Duração de cada período

L^{dia} Limite diário de condução

L^{ext} Extensão diária permitida

L^{sem} Limite semanal de condução

L^{14d} Limite de condução em 14 dias

R_{dia} Descanso diário normal

$R_{\text{dia-red}}$ Descanso diário reduzido

R_{sem} Descanso semanal

P_{pausa} Limite de condução contínua antes da pausa

Sumário

1	INTRODUÇÃO	15
1.1	Contextualização	15
1.2	Problema da Pesquisa	16
1.3	Questão da Pesquisa	17
1.4	Objetivo Geral	17
1.5	Objetivos Específicos	17
1.6	Justificativa	18
1.7	Estrutura da Dissertação	18
2	REVISÃO DA LITERATURA	20
2.1	Introdução	20
2.2	Transporte Rodoviário Europeu e as Restrições Regulatórias	20
2.3	Problemas de Escalonamento e Abordagens Estruturadas	21
2.4	Modelagem Matemática com Programação Linear Inteira	21
2.5	Metaheurísticas e Técnicas Avançadas de Otimização	22
2.6	O Uso de Solvers Modernos: OR-Tools e CP-SAT	22
2.7	Modelos de Apoio à Decisão e Multicritério	23
2.8	Trabalhos Correlatos sobre Escalonamento e Logística	23
2.9	Síntese da Revisão	24
3	BASES LEGAIS E NORMATIVAS DO ESCALONAMENTO DE MOTORISTAS	27
3.1	Introdução	27
3.2	Regulamento (CE) n.º 561/2006	27
3.2.1	Limites de Condução	28
3.2.2	Pausas Obrigatórias	28
3.2.3	Descanso Diário	28
3.2.4	Descanso Semanal	28
3.2.5	Implicações para o modelo matemático	29
3.3	Diretiva 2002/15/CE	29
3.4	Regulamento (UE) n.º 165/2014	30
3.5	Síntese das Bases Legais	30
3.6	Relação com o Modelo Matemático	31
3.7	Considerações Finais	32
4	METODOLOGIA	33

4.1	Introdução	33
4.2	Estrutura Geral da Pesquisa	33
4.3	Etapa 1: Estudo da Legislação Europeia	34
4.4	Etapa 2: Formulação Matemática	34
4.5	Etapa 3: Implementação Computacional	35
4.6	Etapa 4: Definição dos Cenários Experimentais	35
4.7	Etapa 5: Análise dos Resultados	35
4.8	Considerações Finais	36
4.8.1	Abordagem heurística e matheurística proposta	36
4.8.1.1	Heurística construtiva para geração de escala inicial	37
4.8.1.2	Matheurística LNS baseada em MILP	39
4.8.1.3	Integração com aprendizagem de máquina	41
4.9	Metodologia de Avaliação das Soluções de Escalonamento	41
4.9.1	Demandas por Período	42
4.9.2	Escalonamento Gerado (Driver Scheduling)	42
4.9.3	Taxa de Cobertura	42
4.9.4	Índice de Sobrecarga	43
4.9.5	Índice de Subutilização	43
4.9.6	Mapa de Calor de Cobertura	43
4.9.7	Mapa de Calor da Margem de Segurança	43
4.9.8	Convergência do LNS	44
4.9.9	Relaxação do LNS	44
4.9.10	Comparação entre Modos: Exato, Heurístico e LNS	44
4.9.11	Matriz de Restrições (Antes e Depois das Operações)	44
5	MODELO MATEMÁTICO	45
5.1	Introdução	45
5.2	Discretização Temporal	45
5.3	Conjuntos e Índices	45
5.4	Parâmetros do Modelo	46
5.5	Variáveis de Decisão	46
5.6	Funções-Objetivo do Modelo	47
5.6.1	Maximização da Resposta à Demanda	47
5.6.2	Minimização do Número Total de Motoristas	48
5.6.3	Observação sobre a Seleção do Critério	48
5.7	Restrições do Modelo	48
5.7.1	Atendimento da Demanda	48
5.7.2	Exclusividade de Estado por Período	49
5.7.3	Acúmulo de Condução	49
5.7.4	Limite Diário de Condução	49

5.7.5	Limite Semanal e Quinzenal	49
5.7.6	Pausa Obrigatória após 4,5 horas	49
5.7.7	Descanso Diário Normal ou Reduzido	50
5.7.8	Descanso Semanal	50
5.7.9	Limite de Extensões Diárias	50
5.7.10	Continuidade da Jornada	50
5.8	Discussão sobre Linearização	50
5.9	Observações Computacionais	51
5.10	Considerações Finais	51
6	IMPLEMENTAÇÃO COMPUTACIONAL	52
6.1	Introdução	52
6.2	Arquitetura Geral da Solução	52
6.3	Interface de Parametrização	53
6.4	Configuração das Restrições Legais	54
6.5	Inserção da Demanda Operacional	55
6.6	Construção da Matriz de Restrições	55
6.7	Execução do Solver	56
6.8	Fluxo Computacional do Simulador	58
6.9	Modos de Resolução e Estratégias de Otimização	58
6.10	Geração de Indicadores e Gráficos	59
6.11	Módulo de Operações Elementares	59
6.12	Extensões Heurísticas e Metaheurísticas	61
6.13	Integração de Aprendizado de Máquina	62
6.14	Geração de Datasets e Treinamento dos Modelos	63
6.15	Visualização Avançada e Análise Iterativa	63
6.16	Posicionamento do Simulador como Plataforma Experimental	63
6.17	Considerações Finais	64
7	RESULTADOS	66
7.1	Introdução	66
7.2	Roteiro Experimental	66
7.3	Configuração Geral dos Cenários	66
7.4	Execução do Solver	67
7.5	Comportamento da Demanda	68
7.6	Comparação Demanda <i>versus</i> Alocação	69
7.7	Indicadores Operacionais	71
7.8	Estrutura da Matriz de Restrições	72
7.9	Comparação entre Modos de Resolução	73
7.10	Operações Elementares e Análise Estrutural	73

7.11	Cenários de Rápida Convergência	74
7.12	Análise Gráfica Avançada	75
7.13	Síntese Geral dos Resultados	76
8	REVISÃO COMPARATIVA DA LITERATURA	77
8.1	Revisao Comparativa da Literatura: Modelos de Escalonamento, Roteirizacao e Conformidade Regulamentar	77
8.1.1	Analise Comparativa Critica	79
8.2	Comparação das Técnicas Desenvolvidas com a Literatura e o Estado da Arte	79
8.2.1	Extensões Recentes do Simulador e Alinhamento com Tendências Atuais . .	81
8.2.2	Heurística Construtiva como Baseline Experimental	81
8.2.3	Large Neighborhood Search com Reotimização via Solver Inteiro	81
8.2.4	Aprendizado de Máquina como Camada Opcional de <i>Guidance</i>	82
8.2.5	Visualização Analítica e Instrumentação Experimental	82
8.3	Pipeline Híbrido de Otimização	84
8.3.1	Heurística Construtiva e Geração da Solução Inicial	84
8.3.2	Atualização de Estado e Consistência Regulamentar	84
8.3.3	Análise Estrutural da Matriz de Restrições	84
8.3.4	Large Neighborhood Search com Reotimização MILP	85
8.3.5	Camada Opcional de Aprendizagem de Máquina	85
8.3.6	Critério de Parada e Solução Final	85
9	CONCLUSAO E TRABALHOS FUTUROS	86
9.1	Considerações Iniciais	86
9.2	Síntese do Trabalho Desenvolvido	86
9.3	Principais Resultados e Evidências Empíricas	87
9.4	Contribuições Científicas Consolidadas	88
9.5	Posicionamento Científico da Dissertação	88
9.6	Limitações Identificadas	89
9.7	Direções para Trabalhos Futuros	89
9.8	Considerações Finais	90
	Referências	91
	Glossário	93

APÊNDICES	95	
APÊNDICE A – FORMULAÇÕES ALTERNATIVAS DO MODELO 96		
A.1	Modelo com Janelas de Tempo Suaves (Soft Time Windows)	96
A.2	Modelo com Agrupamento de Motoristas por Categoria	96
A.3	Modelo com Múltiplos Períodos (Planeamento Semanal)	97
A.4	Modelo com Preferências do Motorista	97
APÊNDICE B – CENÁRIOS DE SIMULAÇÃO E PARÂMETROS UTILIZADOS 98		
B.1	Descrição Geral dos Cenários	98
B.2	Configurações do Solver (CP-SAT / OR-Tools)	98
B.3	Restrições Consideradas	99
B.4	Dados Sintéticos Gerados	99
APÊNDICE C – ESTRUTURA DE ENTRADAS E SAÍDAS DO SOLVER 100		
C.1	Entradas do Modelo	100
C.2	Saídas do Modelo	100
APÊNDICE D – PROCESSO DE GERAÇÃO DE DADOS SINTÉTICOS 102		
D.1	Estratégia de Geração	102
D.2	Validação dos Dados	102
APÊNDICE E – MODELO MATEMÁTICO COMPLETO 104		
E.1	Variáveis de Decisão	104
E.2	Função Objetivo	104
E.3	Restrições	104
E.3.1	Atribuição Única	104
E.3.2	Límite de Condução Diária	104
E.3.3	Janelas de Tempo	104
E.3.4	Compatibilidade Motorista–Tarefa	105
E.3.5	Acúmulo Semanal	105
E.4	Modelo Final	105
ANEXOS	106	
ANEXO A – RESUMO ESTRUTURADO DO REGULAMENTO (CE) N.º 561/2006 107		

ANEXO B – GLOSSÁRIO TÉCNICO	109
ANEXO C – FLUXO GERAL DA APLICAÇÃO	110

1 Introdução

Os elementos pré-textuais apresentados a seguir têm como finalidade orientar o leitor acerca da estrutura gráfica, simbólica e organizacional desta dissertação, dedicada ao desenvolvimento de um modelo de Programação Linear Inteira para o escalonamento de motoristas sob o Regulamento (CE) n.º 561/2006.

Ao longo do trabalho, diferentes tipos de representações são utilizados para apoiar a formulação matemática, a implementação computacional e a análise dos resultados. A Lista de Ilustrações reúne as figuras que apresentam, entre outros elementos, a interface do sistema construído, a estrutura da matriz de restrições e os gráficos derivados das simulações realizadas. As listas de Quadros e Tabelas são mantidas para conformidade com as normas acadêmicas, mesmo que o texto não contenha quadros ou tabelas formais.

A Lista de Abreviaturas e Siglas consolida os termos técnicos utilizados no campo da Pesquisa Operacional, da otimização e da legislação europeia aplicada ao transporte rodoviário, facilitando o acompanhamento da leitura. A Lista de Símbolos organiza as notações matemáticas empregadas na formulação do modelo, servindo como referência rápida para os capítulos que apresentam o formalismo teórico.

Por fim, o Sumário estrutura a organização geral do documento, permitindo uma navegação clara entre as seções que tratam desde o contexto e fundamentação teórica até os experimentos, resultados e conclusões finais. Esses elementos auxiliam na compreensão integral da pesquisa, conferindo ao trabalho maior clareza, rigor e acessibilidade.

1.1 Contextualização

O transporte rodoviário de cargas é um dos pilares da economia europeia, assegurando fluxos contínuos de bens e garantindo a integração logística entre os Estados-Membros. Nesse contexto, os motoristas profissionais constituem o recurso humano central da operação, tornando a gestão das suas jornadas um fator crítico para a eficiência, segurança e conformidade legal do setor.

Mais recentemente, abordagens híbridas que combinam otimização exata, heurísticas e, de forma complementar, técnicas de aprendizado de máquina têm sido exploradas como forma de acelerar a convergência e melhorar a qualidade das soluções iniciais, especialmente em problemas de grande escala e natureza temporal.

A União Europeia possui uma das regulamentações mais rigorosas do mundo no que diz respeito aos tempos de condução, pausas e períodos de descanso dos motoristas. O *Regulamento (CE) n.º 561/2006*, complementado pela *Diretiva 2002/15/CE* e pelo

Regulamento (UE) n.º 165/2014, estabelece limites estritos que visam proteger a saúde dos motoristas, prevenir acidentes e promover condições de trabalho seguras e humanizadas ([Union, 2006, 2002, 2014](#)). Essas normas afetam diretamente a forma como as escalas são construídas, exigindo atenção constante ao cumprimento de requisitos diários, semanais e quinzenais.

Entretanto, na prática, a elaboração manual de escalas é altamente propensa a erros, especialmente em operações de grande escala, caracterizadas por alta variabilidade de demanda, múltiplas janelas temporais, restrições acumulativas e dependências temporais entre os períodos de trabalho. Pequenas decisões tomadas em um período podem invalidar toda a escala futura, resultando em infrações legais, custos adicionais e riscos operacionais.

Nesse cenário, abordagens matemáticas baseadas em Programação Linear Inteira (PLI) emergem como ferramentas adequadas para formalizar e resolver problemas de escalonamento, oferecendo rigor, precisão e capacidade de lidar com múltiplas restrições simultaneamente ([Hillier; Lieberman, 2015](#)), ([Taha, 2017](#)), ([Nemhauser; Wolsey, 1988](#)). De forma complementar, solvers modernos como o OR-Tools disponibilizam recursos computacionais avançados que permitem resolver modelos inteiros de grande escala em tempos reduzidos ([Google, 2025](#)).

Além do modelo de otimização em si, esta pesquisa também materializa um ambiente de simulação e análise, no qual o usuário pode configurar parâmetros operacionais, habilitar/desabilitar restrições legais e observar, de forma visual, como tais escolhas impactam a viabilidade, o número de motoristas necessários e a qualidade da cobertura da demanda. Esse ambiente inclui métricas e gráficos de diagnóstico (por exemplo, cobertura por slot, subutilização, sobrecarga, risco e estabilidade temporal), bem como análises estruturais do modelo por meio de visualizações de matrizes e medidas como densidade, oferecendo suporte à interpretação do comportamento do solver e à validação do modelo.

Esta dissertação insere-se nesse contexto, propondo um modelo completo de PLI para o escalonamento de motoristas em conformidade com o Regulamento (CE) n.º 561/2006, apoiado por uma implementação computacional interativa e orientada à simulação de cenários operacionais. Parte desse desenvolvimento foi previamente publicada em ([Moreira; Pinheiro; Araujo, 2025](#)), que fundamenta e inspira a formulação aprofundada apresentada nesta pesquisa.

1.2 Problema da Pesquisa

O problema desta dissertação consiste em determinar como alojar motoristas ao longo de um horizonte temporal discretizado, de forma a:

- atender à demanda operacional por período;

- respeitar integralmente o Regulamento (CE) n.º 561/2006;
- minimizar o número total de motoristas utilizados;
- garantir escalas contínuas, regulares e operacionalmente viáveis;
- permitir análises de sensibilidade para diferentes cenários.

Adicionalmente, quando o modelo exato se torna computacionalmente custoso em instâncias maiores, torna-se relevante investigar estratégias de busca e melhoria incremental de soluções, incorporadas no próprio simulador, tais como heurísticas construtivas e busca em grandes vizinhanças, preservando a conformidade legal e permitindo análises comparativas entre modos de resolução.

Trata-se de um contexto em que as restrições acumulativas sobre condução diária, semanal e quinzenal; à necessidade de pausas obrigatórias; à variabilidade da demanda; às dependências temporais entre períodos sucessivos; e ao grande número de combinações possíveis entre motoristas e períodos.

1.3 Questão da Pesquisa

A pesquisa é guiada pelo seguinte contexto:

Como formular e implementar um modelo de Programação Linear Inteira capaz de gerar escalas de motoristas legalmente válidas, operacionalmente eficientes e alinhadas ao Regulamento (CE) n.º 561/2006?

1.4 Objetivo Geral

Desenvolver, implementar e avaliar um modelo de Programação Linear Inteira para o escalonamento de motoristas no transporte rodoviário europeu, assegurando conformidade com o Regulamento (CE) n.º 561/2006 e promovendo eficiência operacional.

1.5 Objetivos Específicos

- Formular matematicamente o problema, definindo variáveis, parâmetros e restrições;
- Representar as exigências legais do Regulamento (CE) n.º 561/2006 sob a forma de inequações lineares;
- Implementar o modelo utilizando Python e o solver OR-Tools;

- Desenvolver uma interface interativa para simulação de cenários e visualização de resultados;
- Validar o modelo em diferentes horizontes (7, 15 e 30 dias);
- Comparar a alocação gerada com a demanda real ou simulada;
- Avaliar métricas de desempenho, como cobertura, subutilização e sobrecarga;
- Analisar a estabilidade do modelo e o impacto das transformações matriciais;
- Implementar e avaliar modos complementares de resolução e melhoria de soluções (por exemplo, geração de solução inicial e busca em vizinhanças), bem como mecanismos de relaxação e registro de histórico de convergência;
- Estruturar rotinas de experimentação com geração de instâncias/datasets e comparação sistemática de desempenho por meio de métricas e visualizações no simulador.

1.6 Justificativa

Esta pesquisa é relevante por diversas razões:

- **Relevância legal e segurança rodoviária:** o descumprimento do Regulamento (CE) n.º 561/2006 implica multas, riscos à vida dos motoristas e impactos negativos na reputação das empresas;
- **Complexidade operacional:** a variabilidade de demanda e a natureza acumulativa das restrições dificultam a elaboração manual de escalas;
- **Avanço científico:** poucos trabalhos integram o Regulamento (CE) n.º 561/2006 a modelos inteiros formais; esta dissertação contribui para preencher essa lacuna, em linha com ([Erdman; Koch, 2022](#)), ([Moreira; Pinheiro; Araujo, 2025](#));
- **Aplicabilidade industrial:** o modelo possui potencial de uso imediato em empresas de transporte, podendo evoluir para soluções SaaS e integração com sistemas telemáticos.

1.7 Estrutura da Dissertação

Esta dissertação está organizada nos seguintes capítulos:

- **Capítulo 1 – Introdução.**

- **Capítulo 2** – Revisão da literatura sobre Programação Linear Inteira, escalonamento, legislação europeia e trabalhos relacionados.
- **Capítulo 3** – Metodologia, detalhando a abordagem e as etapas da pesquisa.
- **Capítulo 4** – Modelo matemático proposto.
- **Capítulo 5** – Implementação computacional.
- **Capítulo 6** – Resultados obtidos e análise.
- **Capítulo 7** – Conclusão e trabalhos futuros.

2 Revisão da Literatura

2.1 Introdução

Este capítulo apresenta o referencial teórico que fundamenta o desenvolvimento do modelo de otimização e do simulador computacional propostos nesta dissertação. A revisão não se limita a uma descrição conceitual da literatura, mas busca estabelecer uma conexão direta entre os trabalhos existentes e as decisões de modelagem, implementação e experimentação adotadas ao longo do trabalho.

São discutidos aspectos relativos ao transporte rodoviário europeu sob regulamentação estrita, à modelagem matemática baseada em Programação Linear Inteira (PLI), aos problemas de escalonamento com janelas temporais móveis, ao uso de metaheurísticas e matheurísticas em problemas de grande porte, bem como à aplicação de solvers modernos capazes de lidar com estruturas temporais complexas.

Adicionalmente, a revisão destaca lacunas existentes na literatura, em especial no que se refere à integração profunda entre conformidade regulatória, granularidade temporal fina, análise estrutural do modelo e experimentação híbrida, aspectos que motivam e justificam a abordagem proposta nesta dissertação.

2.2 Transporte Rodoviário Europeu e as Restrições Regulatórias

O transporte rodoviário de cargas é um dos pilares da cadeia logística europeia, responsável pela circulação de grande parte dos produtos entre países-membros. Para garantir segurança e padronização operacional, o Regulamento (CE) n.º 561/2006 estabelece limites sobre o tempo de condução, pausas e períodos de descanso. Essas regras impõem uma estrutura de dependências temporais dinâmicas ao planejamento operacional das empresas transportadoras.

Modelos tradicionais de planejamento muitas vezes não incorporam de forma explícita essas janelas móveis — diárias, semanais e quinzenais que determinam limites máximos de condução contínua, acumulada e condições específicas de descanso, o que resulta em formulações incapazes de capturar adequadamente o acoplamento temporal e a complexidade combinatória induzida pela regulamentação. A literatura identifica esse tipo de problema como intrinsecamente complexo, dadas as dependências entre períodos consecutivos ([Pillac; Guéret; Medaglia, 2013](#)).

Assim, qualquer solução computacional voltada ao escalonamento de motoristas deve considerar esse conjunto de restrições regulatórias, sob pena de gerar soluções inviáveis

e não conformes. Modelos robustos precisam integrar, além das condições operacionais, os elementos legais que afetam diretamente o comportamento das variáveis de decisão.

Do ponto de vista da modelagem matemática, essas regras introduzem dependências temporais não locais, uma vez que o cumprimento de uma restrição em determinado período depende do histórico acumulado em janelas móveis de 24 horas, 7 dias e 14 dias. Tal característica diferencia o escalonamento de motoristas de problemas clássicos de alocação de pessoal, exigindo formulações capazes de capturar memória temporal e reinícios condicionais, frequentemente por meio de variáveis auxiliares e restrições acumulativas.

2.3 Problemas de Escalonamento e Abordagens Estruturadas

O escalonamento de motoristas pertence à família dos problemas de escalonamento de pessoal (*workforce scheduling*), que envolvem a alocação de indivíduos a atividades ao longo do tempo, respeitando requisitos de capacidade, janelas temporais e restrições operacionais. Tais problemas são tipicamente formulados como modelos inteiros binários, dada a necessidade de representar decisões discretas de trabalhar ou descansar.

Estudos clássicos mostram a adequação da PLI para representar essas estruturas, sobretudo quando associadas a restrições temporais dependentes, como no caso de motoristas profissionais, demonstrando como decisões temporais de alocação podem ser representadas por matrizes binárias, combinadas a restrições lógicas e acumulativas.

Além disso, (Taylor, 2019) destaca que problemas de escalonamento frequentemente são NP-difíceis e exigem formulações que lidem com alta dimensionalidade. Restrições de janelas móveis, comuns em regulamentações como a europeia, exigem especial atenção para evitar inconsistências ou sobrecargas de condução.

No contexto específico do escalonamento de motoristas, a formulação em PLI frequentemente assume uma estrutura matricial binária do tipo motorista \times período, o que leva a modelos de alta dimensionalidade quando se adota granularidade temporal fina. Essa característica impõe desafios computacionais relevantes, ao mesmo tempo em que permite uma representação precisa das regras legais e operacionais, desde que acompanhada por técnicas adequadas de linearização e decomposição temporal.

2.4 Modelagem Matemática com Programação Linear Inteira

Ao longo deste trabalho, os termos Programação Linear Inteira (PLI), Integer Linear Programming (ILP) e Mixed Integer Linear Programming (MILP) são utilizados de forma intercambiável, conforme a convenção adotada na literatura internacional.

A Programação Linear Inteira (PLI) é amplamente utilizada para modelar proble-

mas combinatórios na área de logística. (Hillier; Lieberman, 2021) destaca que variáveis binárias e restrições lineares permitem representar condições complexas como limites de tempo, dependências condicionais e estruturas acumulativas – características essenciais no escalonamento sob regulamentação.

Modelos híbridos envolvendo ILP e heurísticas também são bastante discutidos na literatura (**pinheiro2011densitycontrol**) e (**pinheiro2008hybridcp**) exploram técnicas de otimização combinando busca local, geração de soluções candidatas e modelos exatos. Essas abordagens mostram que, embora métodos exatos proporcionem precisão, heurísticas podem auxiliar na redução do espaço de busca e no aprimoramento da solução inicial.

No contexto do transporte, (**pinheiro2007transporte**) aplica modelos matemáticos ao problema de otimização de transporte de passageiros, destacando o papel da modelagem formal no atendimento a requisitos operacionais e normativos.

2.5 Metaheurísticas e Técnicas Avançadas de Otimização

Modelos baseados em metaheurísticas são particularmente utilizados para problemas de corte, empacotamento, roteamento e escalonamento. Trabalhos como os de (**pinheiro2016nesting**), (**pinheiro2014parallelbrkga**), (**pinheiro2020biasedrkga**) e (**pinheiro2019twophase**) demonstram que algoritmos híbridos, como *pinheiro2016Nesting* e técnicas de geração e busca, são eficazes em problemas estruturados com alta combinatorialidade.

Essas contribuições evidenciam a relevância de técnicas híbridas para solucionar problemas que, assim como o escalonamento de motoristas, possuem espaços de busca muito grandes.

Entre as abordagens metaheurísticas, o método de *Large Neighborhood Search* (LNS) destaca-se como particularmente adequado para problemas de escalonamento com forte acoplamento temporal. O LNS permite a destruição e reconstrução parcial da solução, mantendo parte das decisões fixas enquanto reotimiza subconjuntos do problema, frequentemente com apoio de modelos exatos. Essa característica torna o LNS especialmente atraente para integração com PLI, configurando abordagens conhecidas como matheurísticas.

2.6 O Uso de Solvers Modernos: OR-Tools e CP-SAT

O CP-SAT, solver principal da Google OR-Tools, é particularmente adequado para problemas com grande quantidade de variáveis binárias e restrições lógicas, como é o caso do escalonamento de motoristas sob regulamentação europeia. Sua capacidade de combinar

técnicas de programação por restrições, propagação de domínios e aprendizado de cláusulas permite lidar de forma eficiente com janelas temporais móveis e restrições acumulativas, superando limitações observadas em solvers puramente baseados em programação linear inteira. Estudos da Google Research ([Google, 2025](#)) mostram que CP-SAT supera solvers tradicionais em várias classes de problemas combinatórios.

O uso de solvers híbridos é consistentemente apoiado pela literatura moderna. A proximidade estrutural entre o escalonamento e problemas como roteamento e corte pode ser observada em trabalhos como ([pinheiro2019vrp](#)), que aplicam algoritmos probabilísticos ao problema de roteamento de veículos com coleta e entrega simultâneas. Esses modelos possuem restrições temporais e operacionais análogas, reforçando a aplicabilidade da OR-Tools ao problema tratado nesta dissertação.

2.7 Modelos de Apoio à Decisão e Multicritério

Embora o foco deste trabalho seja um modelo matemático baseado em PLI, modelos multicritério (MCDA) fornecem fundamentação adicional para o processo decisório em ambientes complexos.

([pinheiro2015projectmanagement](#)) discute o uso da Análise Verbal de Decisão (VDA) em ambientes de múltiplos critérios, reforçando a importância de métodos estruturados. Já ([pinheiro2016itinfra](#)) e ([pinheiro2008production](#)) demonstram como abordagens multicritério podem apoiar decisões em contextos onde múltiplos fatores operacionais coexistem.

Essas contribuições mostram que, embora a solução proposta nesta dissertação utilize um modelo matemático exato, técnicas complementares podem ampliar as perspectivas de análise situacional e avaliação de alternativas.

2.8 Trabalhos Correlatos sobre Escalonamento e Logística

Há uma diversidade de estudos aplicados ao escalonamento e otimização no contexto de logística. ([Moreira; Pinheiro; Araujo, 2024](#)) aborda minimização de custos logísticos em empresas de transporte rodoviário, enquanto ([Moreira; Pinheiro; Araujo, 2025](#)) apresenta um modelo preliminar para escalonamento de motoristas sob regulamentação.

No campo de scheduling, trabalhos como ([pinheiro2005jornal](#)) e ([pinheiro2004timetabling](#)) demonstram estruturas formais que são semelhantes àquelas utilizadas no presente estudo: variáveis binárias, controle de janelas e dependências temporais.

Tais contribuições reforçam o espaço de pesquisa e motivam o desenvolvimento de modelos cada vez mais completos e alinhados às necessidades reais do setor.

2.9 Síntese da Revisão

A revisão da literatura realizada neste capítulo evidencia que o escalonamento de motoristas no transporte rodoviário europeu constitui um problema de elevada complexidade estrutural, decorrente da interação entre requisitos operacionais, restrições legais rigorosas e dependências temporais acumulativas. Diferentemente de problemas clássicos de alocação de recursos, o escalonamento sob o Regulamento (CE) n.º 561/2006 impõe janelas móveis de decisão (24 horas, 7 dias e 14 dias), limites acumulados de condução e regras condicionais de descanso, o que torna inadequadas abordagens simplificadas ou puramente heurísticas sem mecanismos explícitos de verificação de conformidade.

A literatura analisada demonstra que a Programação Linear Inteira (PLI) é uma das abordagens mais adequadas para representar formalmente esse tipo de problema, uma vez que permite modelar decisões discretas, restrições acumulativas e vínculos lógicos de forma rigorosa. Trabalhos clássicos e contemporâneos confirmam que modelos baseados em variáveis binárias, quando corretamente formulados, conseguem capturar com precisão regras regulatórias complexas, desde que acompanhados de técnicas adequadas de discretização temporal e linearização.

Entretanto, também se observa que modelos exatos, quando aplicados isoladamente, podem enfrentar limitações de escalabilidade em instâncias de grande porte ou quando submetidos a demandas altamente variáveis. Por esse motivo, a literatura recente aponta uma tendência clara em direção a abordagens híbridas, que combinam modelos matemáticos exatos com heurísticas construtivas, métodos de busca em grandes vizinhanças (*Large Neighborhood Search – LNS*) e estratégias de decomposição ou relaxação. Essas abordagens permitem explorar o espaço de soluções de forma mais eficiente, mantendo a consistência com o modelo regulatório subjacente.

Os estudos revisados também destacam o papel crescente de solvers modernos, como o CP-SAT da biblioteca OR-Tools, que combinam técnicas de programação por restrições, satisfação booleana e otimização inteira. A capacidade desses solvers de lidar com modelos de grande dimensão, fortemente acoplados no tempo e com estruturas matriciais esparsas, torna-os particularmente adequados para problemas de escalonamento sob regulamentação europeia.

Além disso, a literatura evidencia que o uso de métricas estruturais — como densidade da matriz de restrições, padrões de esparsidade e comportamento temporal das variáveis — pode fornecer informações valiosas sobre a dificuldade do problema e o desempenho esperado do solver. No entanto, observa-se que poucos trabalhos exploram explicitamente essas métricas como instrumentos de diagnóstico, pré-processamento ou orientação heurística, o que revela uma lacuna relevante no estado da arte.

Outro aspecto emergente identificado na revisão é a incorporação de técnicas de

aprendizado de máquina como mecanismos auxiliares de apoio à decisão em problemas de otimização combinatória. Embora ainda incipiente no contexto do escalonamento de motoristas sob regulamentação, essa linha de pesquisa sugere que modelos supervisionados podem ser utilizados para estimar a qualidade de decisões locais, priorizar regiões promissoras do espaço de busca ou acelerar processos heurísticos, desde que integrados de forma não intrusiva e com garantias de reproduzibilidade.

De forma geral, a literatura revela um cenário fragmentado: por um lado, modelos de roteirização e VRP focam majoritariamente em aspectos espaciais e de custo; por outro, modelos regulatórios concentram-se nas restrições temporais, frequentemente sem integração com estratégias avançadas de busca ou análise estrutural. Poucos trabalhos apresentam uma abordagem integrada que combine: (i) formulação exata completa das restrições legais, (ii) mecanismos heurísticos e matheurísticos de melhoria, (iii) instrumentação analítica do comportamento do modelo e (iv) suporte à experimentação sistemática.

Nesse contexto, a presente dissertação posiciona-se como uma contribuição que busca integrar esses diferentes eixos, utilizando a Programação Linear Inteira como núcleo formal, complementada por heurísticas construtivas, métodos LNS, análise estrutural da matriz de restrições e, de forma opcional, mecanismos de aprendizado supervisionado. Essa integração visa não apenas resolver instâncias específicas, mas também oferecer uma plataforma experimental capaz de apoiar análises comparativas, estudos de sensibilidade e investigações científicas mais amplas sobre o escalonamento de motoristas sob restrições legais complexas.

A partir dessa síntese, estabelece-se o arcabouço conceitual e metodológico que fundamenta a formulação do modelo matemático, da implementação computacional e do desenho experimental apresentados nos capítulos subsequentes.

Os resultados desta comparação indicam que o simulador implementa um conjunto de técnicas inéditas, particularmente relevantes pela natureza matricial das operações e pelo uso de métricas estruturais (como densidade) para guiar processos heurísticos e matheurísticos. Essa abordagem não foi identificada na literatura revisada, nem no artigo comparativo utilizado como referência, caracterizando assim uma contribuição original dentro do escopo de escalonamento de motoristas.

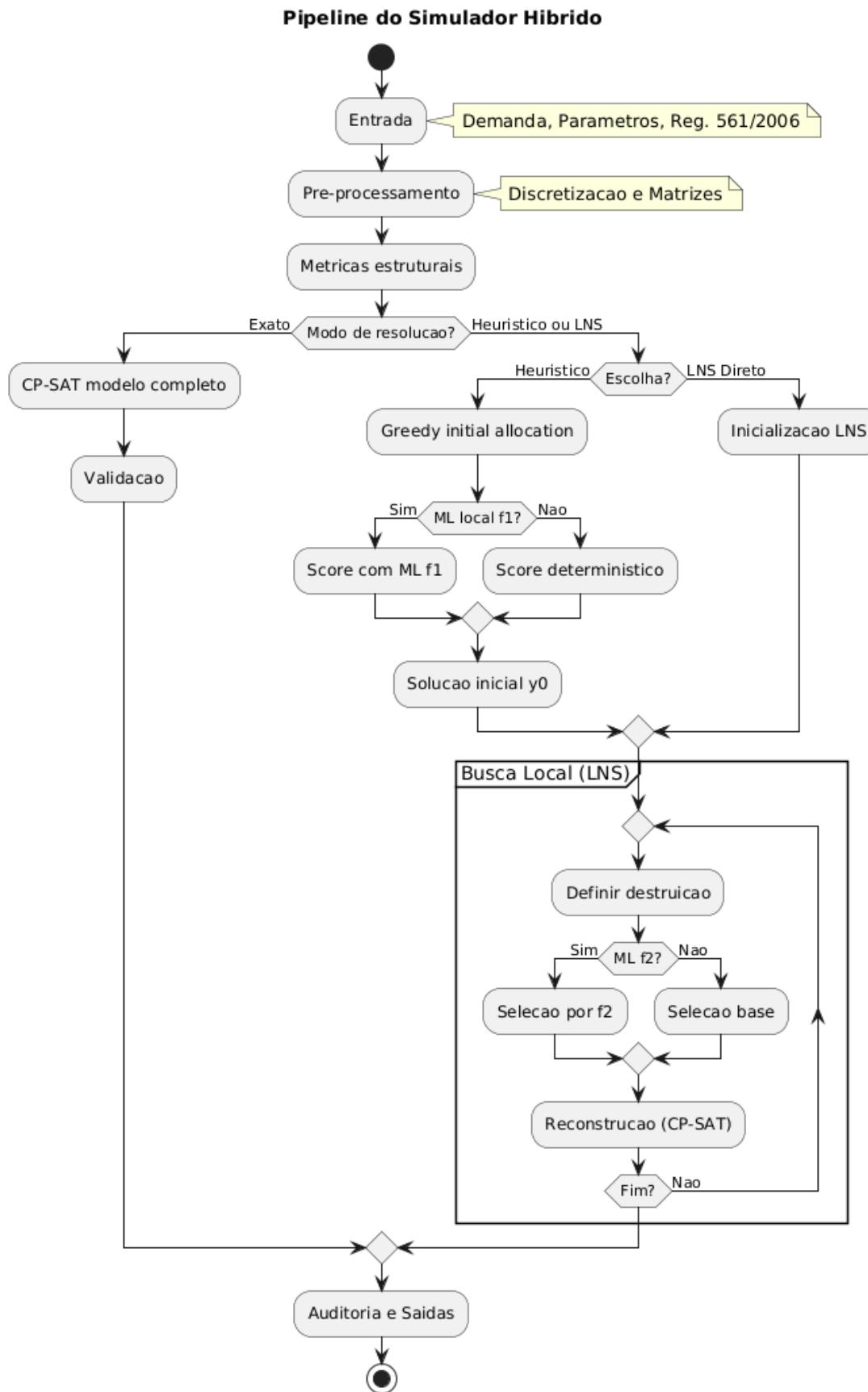


Figura 1 – Pipeline do simulador híbrido proposto: geração de solução inicial por heurística (com ML opcional f_1), melhoria iterativa por LNS (com seleção de vizinhanças opcional f_2), reconstrução local via CP-SAT/MILP com fixação parcial e validação final de conformidade com o Regulamento (CE) n.º 561/2006.

3 Bases Legais e Normativas do Escalonamento de Motoristas

3.1 Introdução

O transporte rodoviário europeu é regulamentado por um conjunto abrangente de normas destinadas a promover a segurança viária, garantir condições adequadas de trabalho aos motoristas e promover a harmonização das práticas entre os Estados-Membros. A formulação do modelo de Programação Linear Inteira (PLI) apresentado nesta dissertação deriva diretamente dessas normas, que estabelecem limites específicos de condução, períodos mínimos de descanso e requisitos tecnológicos relacionados ao registo de atividades.

Além de definir limites normativos, esses diplomas legais impõem uma estrutura temporal altamente acoplada, caracterizada por janelas móveis, restrições acumulativas e dependências entre períodos consecutivos. Tais características tornam o escalonamento de motoristas um problema estruturalmente complexo do ponto de vista computacional, demandando formulações matemáticas capazes de capturar não apenas estados locais, mas também históricos de condução e descanso. Essa complexidade normativa é um dos principais motivadores para a adoção de Programação Linear Inteira, bem como para a exploração de estratégias complementares de resolução e melhoria de soluções, conforme desenvolvido nos capítulos subsequentes.

Este capítulo tem por objetivo apresentar uma síntese estruturada das principais bases legais aplicáveis ao escalonamento de motoristas na União Europeia, com ênfase no Regulamento (CE) n.º 561/2006, na Diretiva 2002/15/CE e no Regulamento (UE) n.º 165/2014. A compreensão desses diplomas jurídicos é fundamental para contextualizar as restrições incorporadas ao modelo matemático discutido no Capítulo 4.

Ressalta-se que as normas aqui descritas não são tratadas apenas como referências teóricas, mas como elementos diretamente observáveis e verificáveis em dados reais de operação. Os parâmetros utilizados ao longo desta dissertação refletem registros efetivos de jornadas, tempos de condução e períodos de descanso, conforme capturados por sistemas operacionais e dispositivos de registro, promovendo aderência prática e sustentação empírica ao modelo proposto.

3.2 Regulamento (CE) n.º 561/2006

O Regulamento (CE) n.º 561/2006 estabelece as regras relativas aos tempos de condução, pausas e períodos de repouso dos motoristas envolvidos em operações de

transporte rodoviário. Seu objetivo principal é melhorar a segurança viária, prevenir a fadiga, reduzir acidentes e garantir condições de trabalho adequadas.

Os principais requisitos incluem:

3.2.1 Limites de Condução

- **Condução diária:** máximo de 9 horas, podendo ser estendido a 10 horas até duas vezes por semana.
- **Condução semanal:** máximo de 56 horas.
- **Condução quinzenal:** máximo de 90 horas em dois períodos consecutivos de 7 dias.

3.2.2 Pausas Obrigatórias

Após um período máximo de 4,5 horas de condução contínua, o motorista deve realizar uma pausa mínima de 45 minutos, podendo ser fracionada em:

- 15 minutos + 30 minutos,
- desde que ambos os períodos ocorram dentro das 4,5 horas.

3.2.3 Descanso Diário

O período de descanso diário pode ser:

- **Normal:** pelo menos 11 horas consecutivas;
- **Reduzido:** pelo menos 9 horas consecutivas, permitido até três vezes entre dois descansos semanais.

3.2.4 Descanso Semanal

- **Regular:** ao menos 45 horas contínuas;
- **Reduzido:** mínimo de 24 horas, com compensação até o final da terceira semana seguinte.

3.2.5 Implicações para o modelo matemático

Essas regras originam diretamente:

- restrições de limites diários, semanais e quinzenais de trabalho;
- restrições acumulativas (janelas móveis);
- restrições de continuidade temporal de descanso;
- necessidade de variáveis auxiliares para extensão diária e pausas.

Do ponto de vista da modelagem, o Regulamento (CE) n.º 561/2006 não pode ser interpretado como um conjunto de restrições isoladas. Seus limites diários, semanais e quinzenais operam simultaneamente e sobrepostos, criando janelas móveis de verificação que se deslocam ao longo do horizonte de planejamento. Essa característica limita a adoção de abordagens puramente locais ou estáticas e justifica a necessidade de variáveis de estado acumulado, mecanismos de reinício após descanso e controle explícito de extensões diárias, todos incorporados ao modelo matemático apresentado no Capítulo 4.

3.3 Diretiva 2002/15/CE

A Diretiva 2002/15/CE trata do *tempo de trabalho* dos trabalhadores móveis. Complementa o Regulamento (CE) n.º 561/2006 ao definir períodos que não são estritamente de condução, mas que fazem parte da jornada de trabalho.

Os principais elementos incluem:

- limite de 48 horas de trabalho semanal (média), podendo atingir 60 horas quando a média de 48 horas for respeitada no período de quatro meses;
- definição de atividades como:
 - carga e descarga;
 - espera operacional;
 - assistência documental e logística;
 - preparação e encerramento de veículo.
- períodos de disponibilidade não contabilizados como trabalho.

Embora o foco principal desta dissertação recaia sobre os tempos de condução regulados pelo Regulamento (CE) n.º 561/2006, os conceitos introduzidos pela Diretiva 2002/15/CE são relevantes para a interpretação dos dados reais utilizados. Em particular,

registros operacionais de jornada frequentemente incluem atividades não classificadas como condução, mas que impactam a disponibilidade do motorista. Assim, mesmo quando não modelados explicitamente como variáveis de decisão, esses tempos influenciam a parametrização do problema e a análise dos resultados obtidos.

Embora esta dissertação concentre-se na parte de condução, os tempos descritos pela Diretiva representam potencial extensão futura do modelo (ver Capítulo 7).

3.4 Regulamento (UE) n.º 165/2014

Este regulamento estabelece as regras relativas ao tacógrafo digital, dispositivo responsável pelo registro eletrônico da condução, pausas, velocidade e distâncias. Sua principal finalidade é assegurar a fiscalização adequada do cumprimento do Regulamento (CE) n.º 561/2006.

Entre seus pontos mais relevantes:

- obrigatoriedade de tacógrafos inteligentes em veículos de transporte internacional;
- registro automático de localização por GNSS;
- salvaguarda e criptografia dos dados de condução;
- integração com sistemas de controle e fiscalização.

No contexto desta dissertação, o Regulamento (UE) n.º 165/2014 estabelece o elo entre o modelo teórico e a realidade operacional. Embora os experimentos utilizem dados simulados, a estrutura do modelo foi concebida de forma compatível com os registros gerados por tacógrafos digitais, permitindo, em princípio, a substituição de dados sintéticos por dados reais. Essa compatibilidade reforça o potencial de aplicabilidade do simulador e viabiliza sua futura integração com sistemas telemáticos e plataformas de gestão de frotas.

Embora não estabeleça limites operacionais adicionais, o Regulamento (UE) n.º 165/2014 define o mecanismo tecnológico que assegura a rastreabilidade, confiabilidade e auditabilidade dos dados utilizados nesta pesquisa. Os registros provenientes de tacógrafos digitais constituem a principal fonte empírica para a validação do cumprimento do Regulamento (CE) n.º 561/2006, reforçando a necessidade de que o modelo matemático esteja alinhado à estrutura e à granularidade dos dados reais disponíveis.

3.5 Síntese das Bases Legais

O conjunto normativo formado pelo Regulamento (CE) n.º 561/2006, pela Diretiva 2002/15/CE e pelo Regulamento (UE) n.º 165/2014 estabelece um arcabouço legal

altamente restritivo e tecnicamente detalhado para o escalonamento de motoristas no transporte rodoviário europeu.

Essas normas impõem:

- limites temporais rígidos e cumulativos;
- janelas móveis de avaliação (24 horas, semana e quinzena);
- forte interdependência entre períodos de condução, pausa e descanso;
- necessidade de monitoramento contínuo e verificável;
- aderência aos registros reais provenientes de tacógrafos digitais.

Do ponto de vista desta dissertação, tais características possuem impacto direto não apenas na formulação matemática, mas também na natureza dos dados utilizados. Os parâmetros empregados refletem situações reais de operação, capturadas por sistemas de registro e gestão, o que exige que o modelo de Programação Linear Inteira seja capaz de reproduzir a lógica normativa observada nos dados empíricos.

3.6 Relação com o Modelo Matemático

Cada restrição normativa apresentada neste capítulo corresponde a um conjunto de inequações no Capítulo 4. Essa abordagem busca assegurar:

- conformidade legal no escopo do modelo;
- capacidade de auditoria;
- rastreabilidade das decisões;
- transparência operacional.

Assim, o modelo matemático não apenas resolve um problema de otimização, mas também se alinha ao quadro regulatório europeu.

Além disso, a utilização de dados reais impõe requisitos adicionais ao modelo, como robustez frente a variações operacionais, consistência temporal e capacidade de reproduzir padrões observados na prática. Dessa forma, o modelo matemático desenvolvido não se limita a satisfazer restrições abstratas, mas busca representar o comportamento observável das jornadas de trabalho sob a legislação europeia.

Dessa forma, o Capítulo 4 apresenta a tradução formal dessas normas em um conjunto estruturado de variáveis, parâmetros, função objetivo e restrições lineares, estabelecendo a base matemática que sustenta toda a implementação computacional e os experimentos apresentados nos capítulos seguintes.

3.7 Considerações Finais

As normativas apresentadas neste capítulo formam a base conceitual e jurídica necessária para compreender o problema de escalonamento de motoristas. Elas estruturam todas as restrições incorporadas ao modelo matemático, justificando a sua formulação e motivando a atenção aos detalhes temporais abordados nos capítulos subsequentes.

4 Metodologia

4.1 Introdução

Este capítulo descreve a abordagem metodológica adotada para o desenvolvimento do modelo de Programação Linear Inteira (PLI) aplicado ao escalonamento de motoristas sob o Regulamento (CE) n.º 561/2006. A metodologia combina rigor matemático, técnicas de Pesquisa Operacional e ferramentas computacionais modernas, buscando reproduzibilidade, precisão e aplicabilidade prática.

A pesquisa pode ser classificada como:

- **Aplicada:** pois aborda um problema concreto do setor de transporte rodoviário.
- **Quantitativa:** devido ao uso de modelagem matemática e experimentação.
- **Metodológica:** por propor um artefato científico (um modelo ILP).
- **Experimental:** pela realização de testes controlados em cenários variados.

Essa abordagem está alinhada aos fundamentos apresentados no ([Hillier; Lieberman, 2015](#)), ([Taha, 2017](#)), ([Nemhauser; Wolsey, 1988](#)), bem como ao trabalho aplicado apresentado em ([Moreira; Pinheiro; Araujo, 2025](#)).

4.2 Estrutura Geral da Pesquisa

A metodologia foi organizada em cinco macroetapas, ilustradas a seguir:

1. Estudo da legislação europeia e identificação das restrições operacionais;
2. Formulação do modelo matemático;
3. Implementação computacional utilizando Python e OR-Tools;
4. Definição de cenários e experimentos;
5. Análise dos resultados e avaliação da conformidade legal.

Cada etapa é detalhada nas seções seguintes.

4.3 Etapa 1: Estudo da Legislação Europeia

Nesta etapa, foi realizada uma análise minuciosa dos sistemática legais que regem a jornada dos motoristas profissionais na União Europeia, incluindo:

- *Regulamento (CE) n.º 561/2006;*
- *Diretiva 2002/15/CE;*
- *Regulamento (UE) n.º 165/2014.*

As restrições extraídas desses documentos foram traduzidas em condições formais, como:

- limites de condução diária, semanal e quinzenal;
- pausas obrigatórias;
- descanso diário normal ou reduzido;
- descanso semanal;
- regras de continuidade temporal.

O desafio principal foi converter regras textuais em inequações lineares, conforme discutido em ([Pillai; Ulmanen, 2019](#)) e explorado experimentalmente em ([Moreira; Pinheiro; Araujo, 2025](#)).

4.4 Etapa 2: Formulação Matemática

Com base nos requisitos legais e operacionais, o problema foi formalizado como um modelo de Programação Linear Inteira (PLI). Os principais elementos definidos foram:

- conjuntos (motoristas, períodos, janelas temporais);
- variáveis de decisão (alocação, descanso, início de jornada, condução acumulada);
- parâmetros (demanda, limites legais, duração dos períodos);
- função objetivo (minimização de motoristas alocados);
- restrições (legais e operacionais).

A formulação foi construída com base em técnicas clássicas de otimização combinatoria ([Nemhauser; Wolsey, 1988](#)), ([Dantzig, 1963](#)), mas adaptada ao contexto legislativo europeu. O modelo é apresentado em detalhes no Capítulo 4, incluindo linearizações e implicações lógicas.

4.5 Etapa 3: Implementação Computacional

A implementação foi realizada em Python, com uso do solver CP-SAT ([Google, 2025](#)), como ferramenta de solução. Essa etapa envolveu:

- criação programática das variáveis do modelo;
- construção da matriz de restrições;
- aplicação das regras legais e dos vínculos sequenciais;
- configuração de parâmetros do solver, conforme critérios experimentais (limites de tempo, cortes, heurísticas);
- desenvolvimento de uma interface interativa em Streamlit;
- geração automática de relatórios, gráficos e indicadores operacionais.

A linguagem Python facilitou o uso de estruturas de dados eficientes, como arranjos NumPy ([Developers, 2025](#)) e ([tabelas Pandas Docs](#)), além de permitir modularidade na construção do modelo.

4.6 Etapa 4: Definição dos Cenários Experimentais

Três tipos de cenário foram definidos para validação:

- **7 dias**: cenário compacto, útil para validar restrições diárias.
- **15 dias**: cenário intermediário, crítico para restrições quinzenais.
- **30 dias**: cenário de escala real, avaliando o comportamento temporal do modelo.

Cada cenário utilizou uma curva de demanda variável, simulando altas e baixas operacionais típicas do setor.

Parâmetros como duração de períodos, restrições legais e tamanho da força de trabalho foram configuráveis via interface.

4.7 Etapa 5: Análise dos Resultados

Após a execução do solver, foram analisados:

- conformidade legal (restrições diárias, semanais e quinzenais);

- cobertura da demanda por período;
- sobrecarga e subutilização;
- estabilidade temporal da alocação;
- tempo computacional e qualidade da solução;
- impacto da estrutura matricial no desempenho do solver.

A avaliação baseou-se em indicadores estatísticos e gráficos, discutidos no Capítulo 6.

4.8 Considerações Finais

A metodologia adotada busca assegurar:

- rigor na formulação matemática;
- consistência na execução computacional;
- flexibilidade para análise de cenários;
- reproduzibilidade científica.

O fluxo metodológico permite integrar legislação, modelagem e computação, resultando em uma solução robusta e aplicável para operações reais no setor de transporte rodoviário europeu.

4.8.1 Abordagem heurística e matheurística proposta

Embora o modelo de programação inteira mista (MILP) formulado neste trabalho permita obter soluções ótimas em instâncias de porte moderado, a sua aplicação direta em cenários de grande escala (muitos motoristas, múltiplos dias e diferentes perfis de demanda) pode tornar-se computacionalmente onerosa. Além disso, em ambientes operacionais dinâmicos, como o transporte rodoviário de mercadorias, nem sempre é necessário obter a solução ótima global; muitas vezes, soluções de boa qualidade obtidas em tempos reduzidos são preferíveis.

Com esse contexto, propõe-se uma abordagem em três camadas: (i) uma heurística construtiva para gerar uma escala inicial factível, (ii) uma matheurística do tipo *Large Neighborhood Search* (LNS), em que subproblemas são resolvidos exatamente pelo modelo MILP/CP-SAT, e (iii) um módulo de apoio baseado em aprendizagem de máquina (*machine learning*) para guiar a seleção de vizinhanças promissoras e refinar decisões locais de alocação.

A heurística construtiva opera em uma linha do tempo discretizada (intervalos de 15 minutos) e, para cada período, atribui motoristas elegíveis de forma gulosa, respeitando as principais restrições regulatórias (condução contínua máxima, pausas mínimas, descanso diário e semanal, limites semanais e quinzenais). Esta fase produz rapidamente uma solução inicial factível, ainda que potencialmente distante do ótimo.

Na segunda camada, aplica-se uma matheurística LNS: a solução corrente é parcialmente destruída em sub-regiões (por exemplo, um dia específico, um subconjunto de motoristas ou janelas temporais críticas com sobrecarga de demanda), e um subproblema restrito é reotimizado com o modelo MILP proposto, sob limite de tempo. Assim, o modelo exato deixa de atuar sobre toda a instância e passa a ser utilizado como um *mecanismo* de melhoria local, combinando a robustez de soluções exatas com a flexibilidade exploratória típica de heurísticas.

Por fim, a terceira camada explora técnicas de aprendizagem de máquina para apoiar a heurística e a matheurística. A partir de instâncias menores, nas quais o MILP consegue atingir a solução ótima em tempo aceitável, constroem-se conjuntos de dados que relacionam padrões de alocação (por motorista, por período e por janela temporal) à qualidade da solução. Modelos supervisionados podem então ser treinados para: (a) atribuir escores heurísticos de preferência a pares motorista-período, auxiliando a etapa construtiva, e (b) estimar a probabilidade de melhoria ao selecionar determinadas vizinhanças na LNS, priorizando regiões da solução com maior potencial de ganho.

Essa arquitetura híbrida (MILP + heurística + LNS + ML) permite, por um lado, preservar o rigor do modelo regulatório desenvolvido e, por outro, oferecer tempos de resposta compatíveis com cenários de apoio à decisão em contexto real, seja em ambiente acadêmico (simulações) ou em produto (como no contexto da plataforma Ottimizia).

4.8.1.1 Heurística construtiva para geração de escala inicial

A seguir apresenta-se a heurística construtiva proposta para geração de uma escala inicial factível.

Algoritmo 1: Heurística construtiva para escalonamento de motoristas

Input: Conjunto de motoristas D ;

Conjunto de períodos discretizados T (intervalos de 15 minutos);

Demandas de motoristas por período $\text{dem}(t)$;

Parâmetros regulatórios (limites de condução e trabalho, pausas e descansos).

Output: Escala inicial $y_{d,t} \in \{0, 1\}$ indicando se o motorista d trabalha no período t .

- 1 Inicializar $y_{d,t} \leftarrow 0$, para todo $d \in D, t \in T$
- 2 Inicializar o estado de cada motorista d (tempo de condução acumulado, tempo de trabalho no dia, semana etc.)
- 3 **foreach** $t \in T$ em ordem cronológica **do**
 - 4 calcular $k \leftarrow$ número de motoristas já alocados em t
 - 5 **if** $k \geq \text{dem}(t)$ **then**
 - 6 **continuar** para o próximo período
 - 7 determinar conjunto de candidatos elegíveis $C_t \subseteq D$ tais que, se escalados em t , não violam nenhuma restrição regulatória (condução máxima, pausa mínima, descanso diário/semanal etc.)
 - 8 **while** $k < \text{dem}(t)$ e $C_t \neq \emptyset$ **do**
 - 9 **foreach** $d \in C_t$ **do**
 - 10 calcular um escore heurístico $\text{score}(d, t)$, por exemplo:
 - menor carga de trabalho acumulada na semana;
 - menor número de dias consecutivos trabalhados;
 - proximidade de completar um bloco consistente de trabalho ou condução.
 - 11 selecionar $d^* \in C_t$ com maior $\text{score}(d^*, t)$
 - 12 definir $y_{d^*,t} \leftarrow 1$
 - 13 atualizar o estado regulatório de d^* (tempo de condução contínua, total diário, semanal etc.)
 - 14 remover d^* de C_t
 - 15 atualizar $k \leftarrow k + 1$
 - 16 **if** $k < \text{dem}(t)$ **then**
 - 17 marcar o período t como *crítico* (demanda não atendida) para uso posterior na LNS
 - 18 **foreach** $d \in D$ **do**
 - 19 pós-processar a escala de d para inserir explicitamente blocos de pausa e descanso, caso ainda não tenham sido definidos de forma clara, garantindo conformidade regulatória final
 - 20 **return** $y_{d,t}$

4.8.1.2 Matheurística LNS baseada em MILP

A heurística construtiva gera uma solução inicial y . Para melhorar essa solução, propõe-se uma matheurística do tipo *Large Neighborhood Search* (LNS), na qual partes da escala são destruídas e reotimizadas por meio do modelo MILP.

Algoritmo 2: LNS matheurística para melhoria da escala

Input: Solução inicial y (escala factível);
 Conjunto de motoristas D , períodos T e demais parâmetros;
 Número máximo de iterações it_{max} ;
 Tempo máximo de resolução do subproblema MILP t_{max}^{MILP} .

Output: Solução melhorada y^{best} .

```

1 Definir função objetivo  $F(y)$  (por exemplo, número de motoristas utilizados,
   soma de horas extras, penalizações por slots não atendidos etc.)
2 Inicializar  $y^{curr} \leftarrow y$  e  $y^{best} \leftarrow y$ 
3 for  $it = 1$  to  $it_{max}$  do
4   selecionar um tipo de vizinhança  $V$  (por exemplo: dia, subconjunto de
      motoristas, janela crítica)
5   if  $V = "dia"$  then
6     escolher um dia  $d_{dia}$  e definir subconjunto de períodos  $T_V \subset T$ 
      correspondentes a esse dia
7     definir subconjunto de motoristas  $D_V \leftarrow D$  (todos os motoristas são
      candidatos para aquele dia)
8   else
9     if  $V = "motoristas"$  then
10       escolher subconjunto de motoristas  $D_V \subset D$  (por exemplo, os mais
          sobrecarregados)
11       definir subconjunto de períodos  $T_V \leftarrow T$  (todos os períodos)
12     if  $V = "janela critica"$  then
13       selecionar subconjunto de períodos  $T_V \subset T$  com slots não atendidos
          ou muito próximos a violações
14       definir subconjunto de motoristas  $D_V \subset D$  que atuam em  $T_V$ 
15   construir subproblema MILP restrito a  $D_V$  e  $T_V$ :
    • variáveis de decisão  $y_{d,t}$  apenas para  $d \in D_V, t \in T_V$ ;
    • manter fixas todas as demais decisões de  $y^{curr}$  fora da vizinhança;
    • preservar todas as restrições regulatórias (diárias, semanais, quinzenais);
    • respeitar consistência com a solução fixa (por exemplo, limites semanais
      acumulados).

```

resolver o subproblema MILP com limite de tempo t_{max}^{MILP} , obtendo uma solução local y^V (se viável)

if foi encontrada solução viável y^V then

```

  construir uma solução candidata  $y^{cand}$  substituindo, em  $y^{curr}$ , as
  decisões de  $D_V \times T_V$  pelas provenientes de  $y^V$ 
  if  $F(y^{cand}) < F(y^{curr})$  then
    atualizar  $y^{curr} \leftarrow y^{cand}$ 
    if  $F(y^{curr}) < F(y^{best})$  then

```

4.8.1.3 Integração com aprendizagem de máquina

Para potencializar a eficiência da LNS, propõe-se o uso de modelos de aprendizagem de máquina para guiar a seleção de vizinhanças e priorizar regiões da solução com maior potencial de melhoria.

Algoritmo 3: Esquema geral de integração com aprendizagem de máquina

Input: Conjunto de instâncias de treinamento;

Soluções ótimas (ou de alta qualidade) obtidas via MILP em instâncias menores;

Heurística construtiva e LNS descritas anteriormente.

Output: Modelo de ML para apoio à decisão na LNS.

1 Fase offline (treinamento):

2 foreach *instância de treinamento* **do**

3 resolver o MILP até a otimalidade ou até um limite de tempo elevado, obtendo solução de referência y^*

4 gerar solução heurística y^H (via heurística construtiva)

5 extrair *features* por período, motorista e janelas temporais (por exemplo: carga acumulada, folgas, distância até limites regulatórios, presença de slots não atendidos)

6 rotular:

- pares (d, t) como bons/ruins com base na diferença entre y^H e y^* ;
- vizinhanças (conjuntos de períodos/motoristas) como promissoras se sua reotimização aproxima y^H de y^* .

7 treinar modelos supervisionados (por exemplo, gradiente reforçado ou *gradient boosting*) para:

- $f_1(d, t) \rightarrow$ estimar a qualidade de atribuir o motorista d ao período t ;
- $f_2(V) \rightarrow$ estimar o ganho esperado de reotimizar uma vizinhança V .

Fase online (uso na LNS):

Na heurística construtiva, utilizar f_1 como componente do escore $\text{score}(d, t)$

Na LNS, ao selecionar vizinhanças, avaliar múltiplas opções V_1, \dots, V_k e priorizar aquelas com maior $f_2(V_i)$

return modelos f_1 e f_2 integrados ao processo heurístico

4.9 Metodologia de Avaliação das Soluções de Escalonamento

A avaliação das soluções geradas pelos métodos implementados no simulador — Programação Inteira (exato), heurística construtiva e método matheurístico LNS — baseia-

se em um conjunto estruturado de indicadores quantitativos, gráficos analíticos e métricas consolidadas na literatura de scheduling e otimização combinatória (Pinedo, 2016), (Talbi, 2009), (Papadimitriou; Steiglitz, 1998), (Moreira; Costa; Amaral, 2025).

Cada indicador descrito nesta seção contribui para a análise multidimensional de desempenho, permitindo verificar simultaneamente: qualidade relativa da solução, estabilidade temporal, segurança operacional, eficiência de cobertura e custo computacional.

As subseções seguintes descrevem formalmente cada métrica, acompanhadas de suas figuras correspondentes.

4.9.1 Demanda por Período

A curva de demanda representa o número de motoristas requeridos em cada intervalo de tempo. Esse gráfico caracteriza a carga operacional e estabelece a referência mínima de cobertura esperada.

Figura 2 – Demanda operacional por período.

Picos de demanda visíveis na [Figura 2](#) implicam maior pressão sobre o algoritmo, enquanto períodos de baixa carga permitem redistribuição estratégica de motoristas.

4.9.2 Escalonamento Gerado (Driver Scheduling)

O gráfico da solução detalha a quantidade de motoristas alocados por período pelo modelo de otimização.

Figura 3 – Motoristas alocados por período na solução gerada.

Comparações entre demanda ([Figura 2](#)) e solução ([Figura 3](#)) apontam aderência, excesso ou déficit operacional.

4.9.3 Taxa de Cobertura

A taxa de cobertura $Coverage_t$ é definida como:

$$Coverage_t = \frac{\text{Motoristas}_t}{\text{Demanda}_t}.$$

Figura 4 – Taxa de cobertura por período.

Valores próximos de 1 indicam aderência adequada, enquanto valores inferiores apontam risco operacional. A [Figura 4](#) sintetiza esse comportamento visualmente.

4.9.4 Índice de Sobrecarga

O índice de sobrecarga é dado por:

$$Overload_t = \frac{\max(0, \text{Motoristas}_t - \text{Demanda}_t)}{\text{Demanda}_t}.$$

Figura 5 – Índice de sobrecarga por período.

A [Figura 5](#) evidencia janelas nas quais há superdimensionamento da escala.

4.9.5 Índice de Subutilização

Definido por:

$$Underload_t = \frac{\max(0, \text{Demanda}_t - \text{Motoristas}_t)}{\text{Motoristas}_t}.$$

Figura 6 – Índice de subutilização por período.

A [Figura 6](#) mostra períodos de déficit, fundamentais para análise de risco operacional.

4.9.6 Mapa de Calor de Cobertura

Figura 7 – Mapa de calor de cobertura por período.

Esse mapa facilita a identificação de padrões estruturais, como clusters de subcobertura.

4.9.7 Mapa de Calor da Margem de Segurança

Figura 8 – Mapa de calor da margem de segurança operacional.

A [Figura 8](#) destaca períodos críticos associados a déficit ou superávit operacional.

4.9.8 Convergência do LNS

Figura 9 – Evolução da função objetivo nas iterações do LNS.

Esse gráfico é fundamental para demonstrar o comportamento da matheurística, evidenciando melhora progressiva ou estagnação.

4.9.9 Relaxação do LNS

Figura 10 – Nível de relaxação aplicado ao longo das iterações do LNS.

Valores elevados sugerem exploração ampla; valores menores refletem intensificação.

4.9.10 Comparação entre Modos: Exato, Heurístico e LNS

Figura 11 – Comparação entre os modos de solução (Exato, Heurístico e LNS).

O radar da [Figura 11](#) sintetiza várias métricas, permitindo comparar os métodos em estabilidade, custo, aderência e eficiência.

4.9.11 Matriz de Restrições (Antes e Depois das Operações)

Figura 12 – Matriz inicial de restrições.

Figura 13 – Matriz final após transformações.

Comparar as Figuras [12](#) e [13](#) revela o impacto direto das operações elementares sobre a densidade e estrutura do modelo.

5 Modelo Matemático

5.1 Introdução

Este capítulo apresenta o modelo de Programação Linear Inteira (PLI) desenvolvido para o escalonamento de motoristas sob o Regulamento (CE) n.º 561/2006. O objetivo do modelo é gerar escalas em conformidade com as restrições legais, eficientes e capazes de atender à demanda operacional por período, incorporando as restrições definidas pela legislação europeia consideradas neste trabalho.

O modelo construído estende a formulação proposta em ([Moreira; Pinheiro; Araujo, 2025](#)), incorporando maior detalhamento temporal, variáveis auxiliares e mecanismos de controle lógico adequados para implementação no solver CP-SAT do OR-Tools.

O modelo foi concebido de forma flexível, permitindo a adoção de diferentes funções objetivo conforme o cenário operacional analisado. Em particular, duas estratégias de otimização são consideradas no simulador desenvolvido: (i) a maximização da resposta à demanda, priorizando a cobertura operacional, e (ii) a minimização do número total de motoristas alocados, priorizando eficiência de recursos. Ambas as abordagens compartilham o mesmo conjunto de variáveis e restrições legais consideradas, diferenciando-se apenas na função objetivo e no tratamento da restrição de atendimento da demanda.

5.2 Discretização Temporal

O horizonte de planejamento é dividido em um conjunto de períodos discretos:

$$T = \{1, 2, \dots, |T|\},$$

onde cada período representa uma janela fixa de tempo com duração:

$$\Delta \text{ (em horas).}$$

Essa discretização permite rastrear a condução acumulada, pausas, descansos e demais restrições temporais do modelo.

5.3 Conjuntos e Índices

- D : conjunto de motoristas ($d \in D$);
- T : conjunto de períodos discretizados ($t \in T$);

- W : janelas agregadas de planejamento:
 - dias,
 - semanas,
 - quinzenas;
- P : parâmetros de demanda por período.

5.4 Parâmetros do Modelo

$demanda_t$: demanda mínima de motoristas no período t ,

$$\begin{aligned} \Delta &: \text{duração de cada período (em horas),} \\ L^{\text{dia}} &= 9 \quad (\text{limite diário de condução}), \\ L^{\text{ext}} &= 10 \quad (\text{extensão diária permitida 2 vezes/semana}), \\ L^{\text{sem}} &= 56 \quad (\text{limite semanal}), \\ L^{14d} &= 90 \quad (\text{limite em 14 dias}), \\ R_{\text{dia}} &= 11 \quad (\text{descanso diário normal}), \\ R_{\text{dia-red}} &= 9 \quad (\text{descanso diário reduzido}), \\ R_{\text{sem}} &= 45 \quad (\text{descanso semanal}), \\ P_{\text{pausa}} &= 4.5 \quad (\text{máximo contínuo de condução antes da pausa}). \end{aligned}$$

Esses parâmetros refletem as exigências legais discutidas em ([Union, 2006](#)), ([Union, 2002](#)).

5.5 Variáveis de Decisão

Variável Principal de Alocação

$$x_{d,t} = \begin{cases} 1, & \text{se o motorista } d \text{ trabalha no período } t, \\ 0, & \text{caso contrário.} \end{cases}$$

Variável de Descanso

$$r_{d,t} = \begin{cases} 1, & \text{se o motorista } d \text{ descansa no período } t, \\ 0, & \text{caso contrário.} \end{cases}$$

Variável de Início de Jornada

$$s_{d,t} = \begin{cases} 1, & \text{se a jornada de } d \text{ inicia em } t, \\ 0, & \text{caso contrário.} \end{cases}$$

Variável de Condução Acumulada

$$h_{d,t} \geq 0.$$

Variável de Extensão Diária

$$z_{d,w} = \begin{cases} 1, & \text{se o motorista utiliza extensão diária na semana } w, \\ 0, & \text{caso contrário.} \end{cases}$$

5.6 Funções-Objetivo do Modelo

O modelo matemático proposto admite duas funções-objetivo alternativas, selecionadas de acordo com o objetivo operacional definido no simulador. Ambas compartilham o mesmo conjunto de variáveis e restrições legais, diferindo apenas no critério de otimização adotado.

A variável u_t utilizada no modo de maximização da resposta à demanda é diretamente explorada nos indicadores apresentados no Capítulo 6, em especial nos índices de subutilização, cobertura e risco operacional. Dessa forma, o modelo matemático estabelece uma ligação direta entre a formulação teórica e os mecanismos analíticos do simulador.

5.6.1 Maximização da Resposta à Demanda

Neste modo, o objetivo do modelo é maximizar o atendimento da demanda operacional ao longo do horizonte de planejamento, penalizando períodos não cobertos.

Define-se a variável auxiliar:

$$u_t \geq 0, \quad u_t \in \mathbb{R} \quad (\text{demanda não atendida no período } t).$$

Sujeita à restrição:

$$\sum_{d \in D} x_{d,t} + u_t = demanda_t, \quad \forall t \in T.$$

A função-objetivo é então dada por:

$$\max Z_1 = \sum_{t \in T} (demanda_t - u_t),$$

ou, de forma equivalente,

$$\min \sum_{t \in T} u_t.$$

Esse critério prioriza a maximização da cobertura, sendo particularmente útil em cenários de alta variabilidade ou restrição severa de recursos.

5.6.2 Minimização do Número Total de Motoristas

Neste modo, o objetivo consiste em minimizar o uso total de motoristas ao longo do horizonte de planejamento, considerando a hipótese de que toda a demanda seja integralmente atendida.

A função-objetivo é definida como:

$$\min Z_2 = \sum_{d \in D} \sum_{t \in T} x_{d,t}.$$

Esse critério busca eficiência operacional e redução de custos de mão de obra, sendo apropriado para cenários estáveis ou de planejamento estratégico.

Observa-se que a minimização do somatório $\sum_{d,t} x_{d,t}$ corresponde à redução do volume total de trabalho alocado. Caso seja desejado minimizar explicitamente o número de motoristas distintos, pode-se introduzir uma variável binária u_d indicando se o motorista d é utilizado no horizonte, com a restrição $x_{d,t} \leq u_d$. No simulador desenvolvido, optou-se pela formulação apresentada, por refletir diretamente o custo operacional associado à utilização efetiva da mão de obra.

5.6.3 Observação sobre a Seleção do Critério

No simulador desenvolvido, o usuário pode selecionar dinamicamente qual função-objetivo será utilizada em cada experimento. Essa escolha não altera a estrutura do modelo, apenas o critério de otimização, permitindo análises comparativas entre estratégias focadas em cobertura ou eficiência operacional.

5.7 Restrições do Modelo

5.7.1 Atendimento da Demanda

O tratamento da demanda depende do modo de otimização selecionado:

- No modo de minimização de motoristas, a demanda constitui uma restrição rígida;
- No modo de maximização da resposta à demanda, admite-se demanda não atendida por meio de variáveis auxiliares.

Essa formulação permite analisar compromissos entre cobertura operacional e disponibilidade de recursos, sem comprometer a conformidade legal no escopo do modelo.

5.7.2 Exclusividade de Estado por Período

Cada motorista deve estar em trabalho ou descanso:

$$x_{d,t} + r_{d,t} = 1, \quad \forall d, t.$$

5.7.3 Acúmulo de Condução

$$h_{d,t} \leq h_{d,t-1} + \Delta x_{d,t}.$$

A dinâmica de reinício do contador de condução após períodos de descanso é implementada por meio de inequações com Big-M, evitando a introdução direta de igualdades não lineares e mantendo a compatibilidade com o solver CP-SAT.

Reinício após descanso (linearização com Big-M):

$$h_{d,t} \leq M(1 - r_{d,t}).$$

5.7.4 Limite Diário de Condução

Para cada motorista e cada dia:

$$\sum_{t \in w_{\text{dia}}} \Delta x_{d,t} \leq L^{\text{dia}} + z_{d,w}.$$

5.7.5 Limite Semanal e Quinzenal

$$\sum_{t \in w_{\text{sem}}} \Delta x_{d,t} \leq L^{\text{sem}},$$

$$\sum_{t \in w_{14d}} \Delta x_{d,t} \leq L^{14d}.$$

5.7.6 Pausa Obrigatória após 4,5 horas

$$h_{d,t} \leq P_{\text{pausa}} + Mr_{d,t}.$$

5.7.7 Descanso Diário Normal ou Reduzido

Para cada janela deslizante de 24 horas:

$$\sum_{t \in \text{janela}_{24h}} \Delta r_{d,t} \geq R_{\text{dia}},$$

ou, caso reduzido:

$$\sum_{t \in \text{janela}_{24h}} \Delta r_{d,t} \geq R_{\text{dia-red}}.$$

5.7.8 Descanso Semanal

$$\sum_{t \in w_{\text{sem}}} \Delta r_{d,t} \geq R_{\text{sem}}.$$

5.7.9 Limite de Extensões Diárias

$$\sum_{w \in \text{semana}} z_{d,w} \leq 2.$$

5.7.10 Continuidade da Jornada

Evita alternância inválida entre estados:

$$x_{d,t-1} - x_{d,t} \leq r_{d,t}.$$

5.8 Discussão sobre Linearização

O Regulamento (CE) n.º 561/2006 contém várias dependências temporais não-lineares, especialmente em regras de:

- descanso diário e semanal,
- condução acumulada,
- pausas após condução contínua,
- limites quinzenais,
- inícios de jornada.

Para permitir resolução via PLI, foi necessário:

- discretizar o horizonte em períodos fixos;

- introduzir variáveis auxiliares ($h_{d,t}, r_{d,t}, s_{d,t}, z_{d,w}$);
- usar técnicas de linearização com Big-M;
- incorporar janelas móveis de 24h, 7 dias e 14 dias.

Esses cuidados permitem a resolução do modelo por solvers inteiros como CP-SAT, que se aproveita de sua estrutura esparsa e temporal ([Google, 2025](#)).

5.9 Observações Computacionais

O modelo completo apresenta:

- dezenas de milhares de variáveis binárias;
- milhares de restrições lineares;
- matriz esparsa com estrutura quase diagonal;
- encadeamento temporal forte.

Ainda assim, o CP-SAT resolve cenários de 7 a 30 dias em poucos segundos, conforme validado no Capítulo 6, o que indica seu potencial de aplicabilidade prática e alinhamento com estudos que aplicam otimização a problemas temporais complexos ([Pillac; Guéret; Medaglia, 2013; Erdman; Koch, 2022](#)).

5.10 Considerações Finais

O modelo matemático apresentado formaliza o escalonamento de motoristas sob a legislação europeia. Com isso, fornece a base científica necessária para a implementação computacional descrita no capítulo seguinte e para as análises empíricas realizadas posteriormente.

Embora o modelo seja resolvido de forma exata, sua estrutura também permite a integração com heurísticas e métodos de aprendizado supervisionado, utilizados posteriormente para geração de soluções iniciais e exploração de vizinhanças, sem alterar a formulação matemática.

6 Implementação Computacional

6.1 Introdução

Este capítulo descreve a implementação computacional do modelo apresentado no Capítulo 4, utilizando a linguagem Python, o solver CP-SAT do OR-Tools e uma interface interativa criada em Streamlit. O objetivo da implementação é proporcionar uma plataforma integrada e flexível para simulação, análise e avaliação de cenários diversos, preservando a estrutura e as restrições legais propostas pelo Regulamento (CE) n.º 561/2006.

A solução foi construída em camadas: interface, modelagem, solver e análise de resultados. Essa abordagem modular facilita manutenção, extensibilidade e integração futura com sistemas corporativos ou plataformas SaaS.

A implementação foi concebida para operar tanto com dados reais quanto com dados sintéticos de demanda. Nos experimentos apresentados nesta dissertação, os dados de demanda utilizados correspondem a dados reais provenientes de operações de transporte rodoviário específicas, previamente tratados e anonimizados, garantindo conformidade ética e confidencialidade operacional.

A possibilidade de geração sintética de demanda permanece disponível no simulador exclusivamente para fins de testes, validação metodológica e experimentação controlada, não sendo utilizada na análise principal dos resultados apresentados neste trabalho.

6.2 Arquitetura Geral da Solução

A arquitetura computacional é composta por quatro módulos principais:

1. **Interface (Streamlit)**: coleta de parâmetros, configuração de cenários e visualização dos resultados.
2. **Modelagem**: construção das variáveis, função objetivo e restrições.
3. **Solver (OR-Tools)**: execução do CP-SAT, retorno de soluções, logs e tempos de execução.
4. **Pós-processamento**: geração de gráficos, métricas e indicadores operacionais.

Essa organização segue recomendações da literatura contemporânea em problemas combinatórios ([Google, 2025](#)).

Além da formulação exata via Programação Linear Inteira, o simulador foi estendido para incorporar estratégias heurísticas e matheurísticas, permitindo a análise comparativa entre diferentes abordagens de resolução. Em particular, foram implementados três modos de solução: (i) solução exata via CP-SAT, (ii) heurística construtiva gulosa para geração de soluções iniciais e (iii) uma abordagem Large Neighborhood Search (LNS), que combina heurísticas com reotimizações locais via solver inteiro. Essa arquitetura híbrida amplia o escopo experimental da plataforma, possibilitando avaliar compromissos entre qualidade da solução e esforço computacional.

Adicionalmente, o simulador incorpora mecanismos de aprendizado de máquina supervisionado com o objetivo de guiar decisões heurísticas. Dois modelos foram introduzidos: um modelo local de atribuição motorista–período, responsável por estimar a atratividade de alocações individuais, e um modelo de avaliação de vizinhanças, utilizado para priorizar regiões promissoras no processo LNS. Esses modelos são treinados a partir de dados sintéticos gerados pelo próprio simulador, garantindo consistência entre o processo de aprendizado e o domínio do problema. Ressalta-se que a ativação do aprendizado de máquina é opcional, sendo o comportamento padrão totalmente determinístico e compatível com a heurística clássica.

Do ponto de vista metodológico, a implementação computacional foi concebida como um simulador híbrido, no qual diferentes estratégias de resolução coexistem e podem ser avaliadas sob um mesmo conjunto de parâmetros e restrições legais. A Programação Linear Inteira constitui o núcleo normativo do sistema, preservando a conformidade jurídica e consistência matemática, enquanto heurísticas, metaheurísticas e modelos de aprendizado de máquina são integrados como mecanismos auxiliares, com o objetivo de ampliar a eficiência computacional e o escopo experimental da plataforma.

Todas as soluções finais permanecem integralmente validadas pelo modelo de Programação Linear Inteira e pelo solver CP-SAT, garantindo conformidade legal e rigor matemático independentemente da ativação ou não do aprendizado de máquina.

6.3 Interface de Parametrização

A interface inicial permite ao usuário configurar:

- número total de períodos;
- granularidade temporal;
- tipo de variáveis (binárias ou relaxadas);
- solver (CP-SAT ou SCIP);

- ativação ou desativação de restrições legais;
- demanda por período.

A Figura 14 apresenta a tela inicial da interface do sistema.

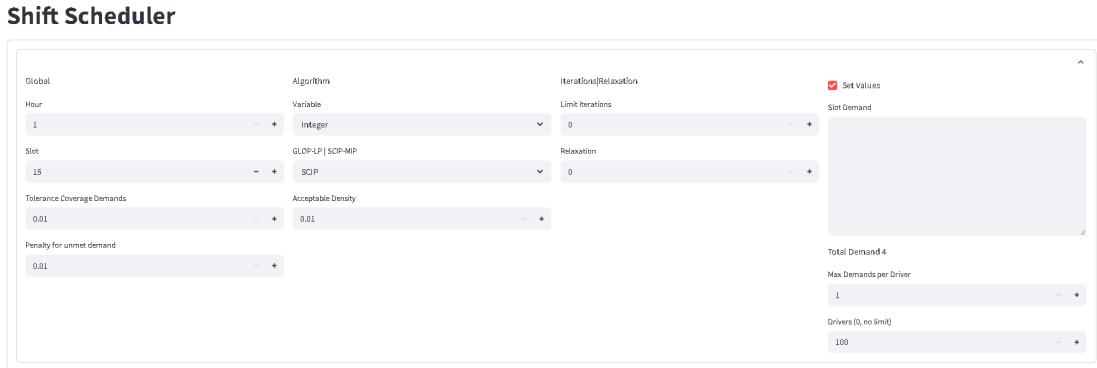


Figura 14 – Interface principal de parametrização da solução.

6.4 Configuração das Restrições Legais

O sistema permite ativar ou desativar individualmente as restrições associadas ao Regulamento (CE) n.º 561/2006, o que possibilita:

- realizar análises comparativas;
- simular cenários educacionais;
- testar hipóteses de flexibilização em ambiente experimental;
- avaliar impacto de restrições específicas.

A Figura 15 ilustra essa funcionalidade.

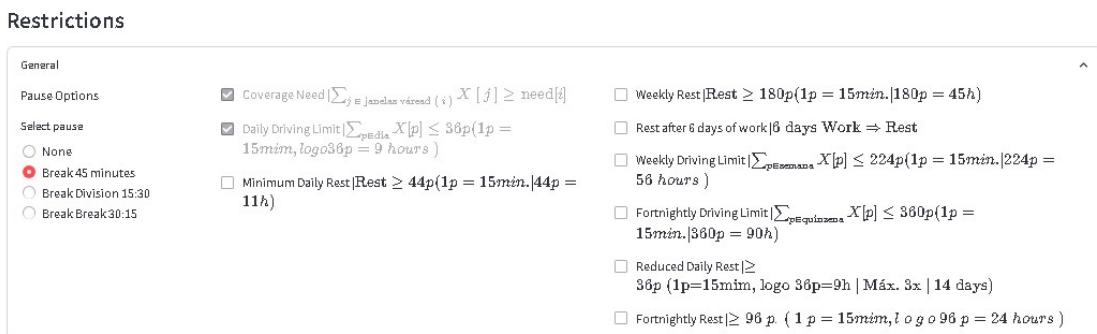


Figura 15 – Seção de ativação das restrições do Regulamento (CE) n.º 561/2006.

6.5 Inserção da Demanda Operacional

A demanda por período pode ser definida:

- manualmente pelo usuário;
- carregada a partir de arquivos externos;
- gerada aleatoriamente para simulação.

A Figura 16 apresenta a demanda utilizada nos experimentos.

Shift Scheduler

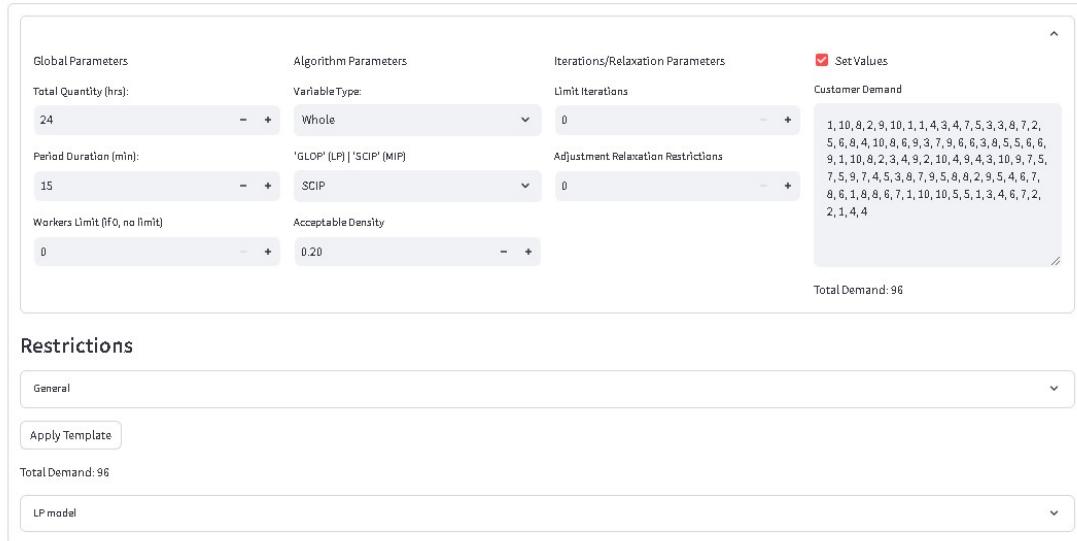


Figura 16 – Distribuição da demanda operacional por período.

6.6 Construção da Matriz de Restrições

A matriz de restrições é um dos componentes centrais da implementação. Ela representa formalmente:

- regras legais;
- vínculos operacionais;
- continuidade temporal;
- acúmulo e reinício de condução;
- períodos de descanso.

As Figuras 17 e 18 mostram, respectivamente, a matriz inicial e a versão final após transformações elementares.

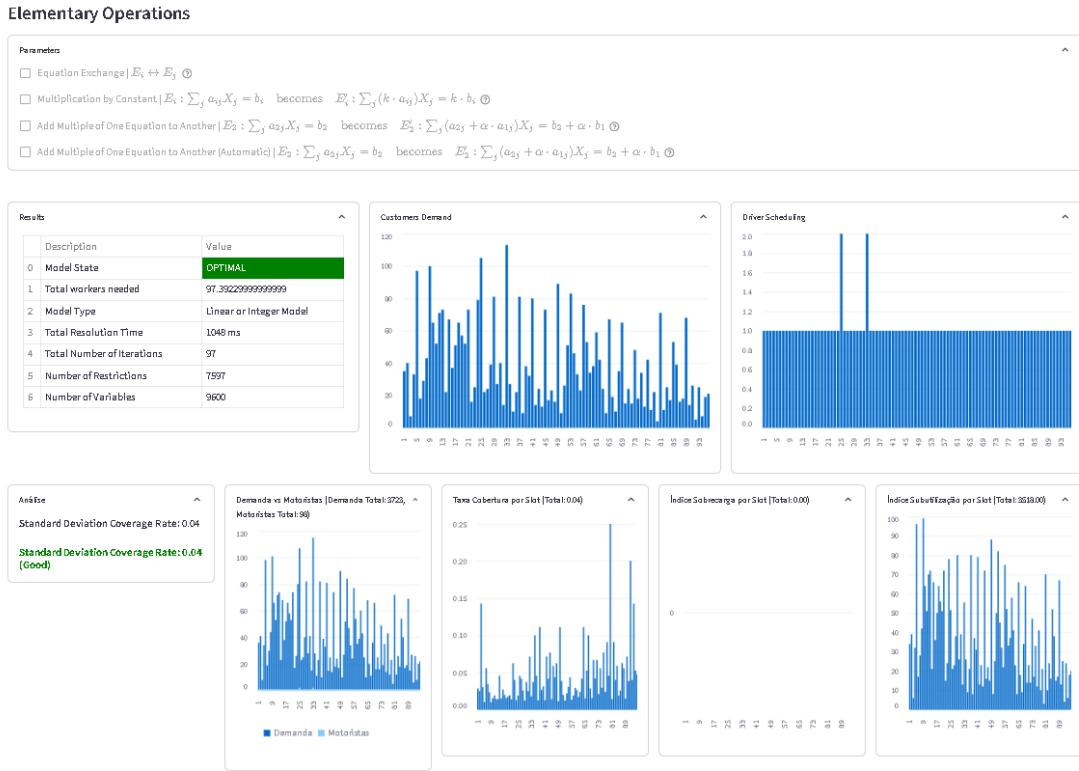


Figura 17 – Matriz de restrições (versão inicial).

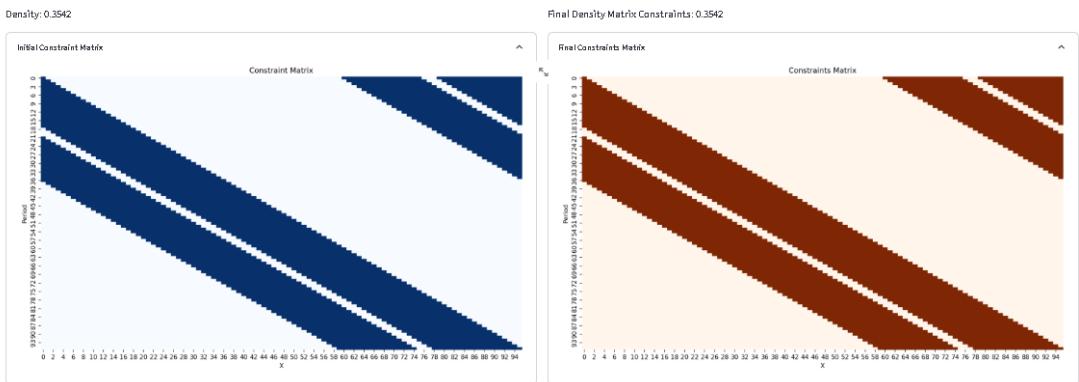


Figura 18 – Matriz de restrições (versão final).

A estrutura esparsa da matriz (entre 15% e 35% de preenchimento) é favorável ao solver, conforme discutido por (Erdman; Koch, 2022).

6.7 Execução do Solver

Após a modelagem, o CP-SAT é executado, retornando:

- estado da solução (ótimo, viável, limite de tempo);
- custo total;
- valores das variáveis;
- métricas de desempenho;
- tempo computacional.

Embora o modelo seja formulado como um problema de Programação Linear Inteira, a resolução é realizada por meio do solver CP-SAT, que combina técnicas de programação por restrições, satisfatibilidade booleana e otimização inteira. Essa abordagem híbrida permite explorar de forma eficiente a estrutura temporal e esparsa do modelo, mitigando limitações observadas de solvers MILP tradicionais.

Um exemplo de execução é mostrado na Figura 19.

Shift Scheduler

The screenshot shows the Shift Scheduler software interface. The top section, "Global", contains settings for "Hour" (24), "Variable" (Integer), "Algorithm" (GLOP-IP | SCIP-MIP), "Iterations/Relaxation" (0), and "Set Demand" (a large list of numbers). Below this are "Tolerance" (0.01), "Acceptable Density" (0.36), and "Penalty for unmet demand" (0.01). The right side shows "Total Demand" (96), "Max Demands per Driver" (1), and "Drivers [0, no Fwd]" (100). The bottom section, "Restrictions", lists various driving and rest constraints. Some checkboxes are checked, such as "Coverage Need" and "Daily Driving Limit". Other options like "Reduced Daily Rest" and "Biweekly Rest" are also present. At the bottom, there are buttons for "Apply Template", "Total Demand 96", and "PL model".

Figura 19 – Execução do solver CP-SAT (solução ótima).

Tempos inferiores a ordem de segundos foram observados nos experimentos realizados mesmo para cenários com milhares de variáveis.

6.8 Fluxo Computacional do Simulador

Do ponto de vista computacional, o simulador segue um pipeline bem definido, composto pelas seguintes etapas sequenciais:

1. leitura e validação dos parâmetros de entrada;
2. construção do modelo matemático (variáveis, restrições e função objetivo);
3. execução do método de resolução selecionado (exato, heurístico ou LNS);
4. pós-processamento da solução obtida;
5. cálculo de indicadores operacionais e geração de visualizações.

Esse pipeline é executado de forma determinística para um conjunto fixo de parâmetros, favorecendo reproduzibilidade dos experimentos realizados.

O fluxo de execução do simulador segue uma sequência bem definida de etapas, independentemente do modo de resolução selecionado. Inicialmente, o usuário configura os parâmetros do problema por meio da interface interativa, incluindo horizonte temporal, demanda, restrições legais e estratégia de solução. Em seguida, o sistema constrói internamente a representação matricial do problema, instanciando variáveis, restrições e a função objetivo conforme o modelo matemático descrito no Capítulo 4.

No modo exato, o modelo completo é submetido diretamente ao solver CP-SAT, que realiza a busca por soluções ótimas respeitando todas as restrições. Nos modos heurístico e LNS, o sistema primeiro gera uma solução inicial viável, que pode ser utilizada isoladamente ou como ponto de partida para processos iterativos de melhoria. No caso do LNS, subconjuntos de períodos são liberados a cada iteração, mantendo o restante da solução fixo, e um problema restrito é resolvido novamente pelo solver inteiro.

Após a obtenção da solução final, o simulador executa uma etapa de pós-processamento, responsável pelo cálculo de indicadores operacionais, métricas de desempenho e geração de visualizações gráficas. Esse fluxo modular permite comparar diferentes estratégias sob condições controladas, garantindo reproduzibilidade e consistência nos experimentos realizados.

6.9 Modos de Resolução e Estratégias de Otimização

É importante distinguir, no contexto do simulador, os conceitos de *modo de resolução* e *modo de otimização*. O modo de resolução define a estratégia algorítmica utilizada (exata, heurística ou matheurística), enquanto o modo de otimização define a função objetivo

adotada pelo modelo matemático (maximização da resposta à demanda ou minimização do número total de motoristas).

Essa separação conceitual permite combinar livremente diferentes estratégias de resolução com diferentes objetivos operacionais, ampliando o escopo experimental da plataforma.

O simulador permite a execução do problema de escalonamento em diferentes modos de resolução. No modo exato, o modelo completo de Programação Linear Inteira é resolvido pelo solver CP-SAT, permitindo a obtenção de soluções ótimas quando o tempo computacional é suficiente. No modo heurístico, uma estratégia construtiva gulosa gera rapidamente uma solução viável, priorizando o atendimento da demanda com menor complexidade computacional. Já no modo LNS (Large Neighborhood Search), o sistema explora iterativamente vizinhanças da solução corrente, liberando subconjuntos de períodos e reotimizando-os via solver inteiro, o que permite melhorias progressivas da solução com controle explícito do esforço computacional.

6.10 Geração de Indicadores e Gráficos

O sistema produz gráficos essenciais para avaliação da qualidade relativada da solução:

- cobertura da demanda;
- sobrecarga;
- subutilização;
- distribuição da quantidade de motoristas ao longo dos períodos;
- desvio padrão da cobertura.

A Figura 20 apresenta alguns desses resultados.

Já a Figura 21 destaca a comparação entre demanda e cobertura.

6.11 Módulo de Operações Elementares

Além de resolver o problema, a aplicação permite aplicar operações elementares à matriz de restrições, possibilitando:

- fins educacionais (ensino de álgebra linear);
- análise estrutural da matriz;

The screenshot displays three main sections of a software interface:

- Cache:** Shows parameters like "Update File" checked and "Initial Density: 0.19%".
- Elementary Operations:** A list of operations with checkboxes: "Equation Exchange", "Multiplication by Constant", "Add Multiple of One Equation to Another", and "Add Multiple of One Equation to Another - Automatic". Below this is a green bar with the message "Great Solution Found!".
- Results:** A table showing various metrics:

	Description	Value
0	Great Solution Found!	
1	Total workers needed	48.0
2	Model Type	Linear or Integer Model
3	Total resolution time	38 ms
4	Total number of iterations	29
5	Number of restrictions	97
6	Number of variables	96

Figura 20 – Indicadores de cobertura, sobrecarga e subutilização.

- diagnósticos estruturais.

Exemplo ilustrado na Figura 22.

Os gráficos analíticos gerados pelo simulador desempenham papel central na avaliação da qualidade das soluções. O gráfico de cobertura da demanda permite verificar, período a período, se o número de motoristas alocados atende ou excede a demanda operacional. O gráfico de margem de segurança evidencia excessos ou déficits de alocação, auxiliando na identificação de sobrecargas ou subutilizações. Já os gráficos temporais de estabilidade e eficiência fornecem uma visão agregada do comportamento da solução ao longo do horizonte, indicando o grau de regularidade das escalas e o nível de aproveitamento da força de trabalho.

No modo LNS, gráficos adicionais apresentam o histórico das iterações, incluindo a evolução do valor da função objetivo e a identificação de melhorias sucessivas. Esses gráficos permitem analisar o processo de convergência do algoritmo, evidenciando como a solução é refinada ao longo das iterações e em quais momentos ocorrem ganhos significativos. Essa visualização é particularmente útil para estudos experimentais, pois permite comparar estratégias de vizinhança, critérios de aceitação e parâmetros de relaxação.

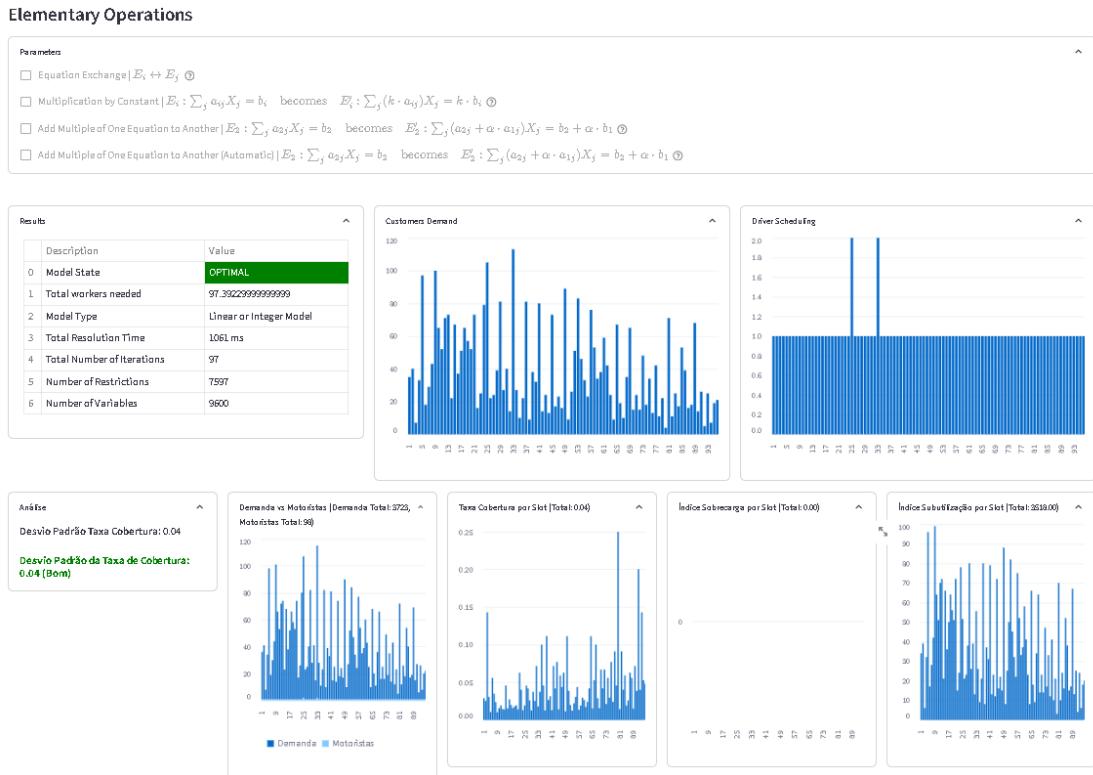


Figura 21 – Demanda comparada à cobertura gerada pelo modelo.

6.12 Extensões Heurísticas e Metaheurísticas

Além da resolução exata via Programação Linear Inteira, o simulador foi estendido para incorporar métodos heurísticos e metaheurísticos, com o objetivo de ampliar a capacidade de experimentação, reduzir tempos computacionais em cenários de maior escala e permitir análises comparativas entre diferentes estratégias de solução.

Inicialmente, foi implementada uma heurística gulosa (*greedy initial allocation*), responsável por gerar soluções viáveis iniciais a partir da demanda por período. Essa heurística prioriza a alocação de motoristas nos períodos de maior demanda, respeitando limites operacionais básicos, e serve tanto como solução independente quanto como ponto de partida para métodos mais avançados.

Em seguida, foi integrado um método de *Large Neighborhood Search* (LNS), caracterizado pela destruição e reconstrução parcial da solução. A cada iteração, subconjuntos de períodos são liberados, mantendo o restante da solução fixo, e um problema restrito é resolvido novamente via solver inteiro. Essa abordagem combina características da otimização exata com a flexibilidade de exploração típica das metaheurísticas.

The screenshot displays a software interface with three main sections:

- Cache**: A panel showing parameters: "Update File" (checked) and "Initial Density: 0.19%".
- Elementary Operations**: A panel listing operations: "Equation Exchange", "Multiplication by Constant", "Add Multiple of One Equation to Another", and "Add Multiple of One Equation to Another - Automatic". A green bar at the bottom indicates "Great Solution Found!".
- Results**: A table showing optimization results:

	Description	Value
0	Great Solution Found!	
1	Total workers needed	48.0
2	Model Type	Linear or Integer Model
3	Total resolution time	38 ms
4	Total number of iterations	29
5	Number of restrictions	97
6	Number of variables	96

Figura 22 – Operações elementares aplicadas à matriz de restrições.

6.13 Integração de Aprendizado de Máquina

O simulador incorpora um módulo experimental de aprendizado de máquina com o objetivo de orientar decisões locais e estratégicas durante o processo de otimização. Essa integração não substitui o modelo matemático, mas atua como um mecanismo de apoio, mantendo compatibilidade com a solução exata.

Foram definidos dois modelos supervisionados distintos. O primeiro, denominado f_1 , atua no nível local, atribuindo escores à decisão de alocar um determinado motorista em um período específico durante a heurística gulosa. O segundo modelo, f_2 , opera em nível agregado, avaliando vizinhanças candidatas no algoritmo LNS, priorizando aquelas com maior potencial de melhoria.

A arquitetura foi concebida de forma resiliente: na ausência de modelos treinados ou do ambiente de aprendizado de máquina, o sistema automaticamente recorre a regras heurísticas clássicas, garantindo que o comportamento do simulador permaneça estável e reproduzível. Dessa forma, o aprendizado de máquina atua como uma camada opcional de inteligência adicional, sem comprometer a validade dos experimentos.

6.14 Geração de Datasets e Treinamento dos Modelos

Para viabilizar a integração com aprendizado de máquina, foi desenvolvido um pipeline automático de geração de dados supervisionados. A partir de múltiplas instâncias sintéticas de demanda, o simulador executa tanto soluções heurísticas quanto soluções ótimas via solver inteiro, registrando decisões e resultados intermediários.

Esses dados são organizados em dois conjuntos distintos: um dataset voltado ao treinamento do modelo f_1 , contendo informações locais sobre motorista, período e contexto operacional; e um segundo dataset para o modelo f_2 , composto por características agregadas das vizinhanças exploradas pelo LNS.

O treinamento dos modelos é realizado externamente ao fluxo principal de otimização, utilizando algoritmos de *gradient boosting* (LightGBM). Os modelos resultantes são então carregados dinamicamente pelo simulador, permitindo a avaliação do impacto do aprendizado supervisionado sobre a qualidade e a estabilidade das soluções geradas.

6.15 Visualização Avançada e Análise Iterativa

Além dos gráficos tradicionais de cobertura e atendimento da demanda, o simulador passou a oferecer visualizações avançadas que permitem uma análise mais profunda do comportamento dos algoritmos ao longo do tempo.

No contexto do LNS, são gerados gráficos que mostram a evolução do valor da função objetivo por iteração, a identificação de melhorias sucessivas e o comportamento do nível de relaxamento aplicado. Esses gráficos permitem avaliar empiricamente a convergência do método e comparar o desempenho entre estratégias exatas, heurísticas e metaheurísticas.

Adicionalmente, foram incluídos mapas de calor e métricas temporais que evidenciam padrões de subutilização, sobrecarga e estabilidade da solução, ampliando a capacidade analítica do simulador e fortalecendo sua utilização tanto para pesquisa quanto para fins educacionais.

6.16 Posicionamento do Simulador como Plataforma Experimental

Com as extensões implementadas, o simulador ultrapassa o papel de uma simples ferramenta de resolução de problemas de Programação Linear Inteira, consolidando-se como uma plataforma experimental híbrida. Ele permite a comparação sistemática entre métodos exatos, heurísticos, metaheurísticos e abordagens híbridas com aprendizado de máquina, sob um mesmo conjunto de restrições legais rigorosas.

Essa flexibilidade torna o sistema particularmente adequado para estudos de sensibilidade, avaliação de escalabilidade e investigação de estratégias avançadas de otimização

aplicada ao transporte rodoviário. Além disso, a arquitetura modular facilita futuras extensões, como modelos estocásticos, integração com dados telemáticos reais e aplicações em ambientes corporativos ou SaaS.

É importante destacar que todas as extensões implementadas — heurísticas, metáheurísticas e aprendizado de máquina — foram concebidas como camadas complementares ao modelo matemático exato, não comprometendo a conformidade normativa no escopo do modelo. Em particular, o uso de aprendizado supervisionado limita-se à priorização de decisões locais e à seleção de vizinhanças promissoras, sem alterar diretamente as restrições ou a função objetivo do problema. Dessa forma, o simulador preserva rigor científico e transparência metodológica, ao mesmo tempo em que amplia significativamente seu potencial experimental.

6.17 Considerações Finais

A implementação computacional desenvolvida mostrou-se:

- eficiente;
- escalável;
- robusta;
- flexível para análise de cenários;
- eficiente em termos computacionais;
- escalável nos cenários avaliados;
- estável sob diferentes configurações;
- flexível para análise de cenários.

O uso combinado de PLI, OR-Tools e Streamlit permite visualizar, interpretar e validar facilmente soluções complexas, oferecendo um ambiente completo para experimentação científica e aplicação prática no setor de transporte rodoviário europeu.

Dessa forma, a implementação computacional transcende a simples resolução de um modelo de Programação Linear Inteira, constituindo uma plataforma experimental completa para o estudo de escalonamento de motoristas sob restrições legais complexas. A combinação de modelagem exata, heurísticas, metáheurísticas, aprendizado de máquina e visualização interativa fornece uma base metodológica para investigações científicas e estudos aplicados avançadas, alinhando-se aos objetivos desta dissertação.

A implementação computacional descrita neste capítulo fornece a base operacional necessária para a realização dos experimentos apresentados no capítulo seguinte. Todos os resultados, indicadores e análises discutidos no Capítulo 7 foram gerados diretamente a partir do simulador aqui descrito, mantendo consistência entre modelo matemático, implementação computacional e análise empírica.

7 Resultados

7.1 Introdução

Este capítulo apresenta os resultados obtidos com a execução do modelo de Programação Linear Inteira (PLI) e da implementação computacional descrita no Capítulo 5. Os experimentos foram realizados para diferentes horizontes temporais (1, 7, 15 e 30 dias), utilizando-se demanda variável e ativação integral das restrições impostas pelo Regulamento (CE) n.º 561/2006.

Os resultados demonstram que o modelo é capaz de gerar escalas em conformidade com as restrições normativas modeladas, eficientes e compatíveis com cenários operacionais representativos de transporte rodoviário, apresentando tempos de execução reduzidos nos cenários avaliados e alta estabilidade temporal.

7.2 Roteiro Experimental

Os resultados apresentados neste capítulo seguem um roteiro experimental estruturado, concebido para avaliar de forma progressiva o comportamento do simulador sob diferentes estratégias de resolução e níveis de complexidade. Esse roteiro foi organizado em fases, permitindo analisar separadamente a viabilidade do modelo matemático, o desempenho computacional, a qualidade das soluções e o impacto de estratégias heurísticas, matheurísticas e orientadas por aprendizado de máquina.

As fases experimentais contemplam: (i) otimização exata via Programação Linear Inteira, (ii) geração de soluções iniciais por heurística gulosa, (iii) refinamento das soluções por Large Neighborhood Search (LNS), (iv) integração de modelos de aprendizado de máquina supervisionado e (v) avaliação comparativa entre os diferentes modos de resolução. Essa abordagem sistemática favorecendo rastreabilidade, reproduzibilidade computacional e rigor metodológico na análise dos resultados.

7.3 Configuração Geral dos Cenários

Os cenários foram configurados conforme os seguintes parâmetros principais:

- solver: CP-SAT (OR-Tools);
- granularidade: 15 minutos por período;
- horizontes avaliados: 1, 7, 15 e 30 dias;

- variável principal: $x_{d,t}$ (binária);
- ativação completa das restrições legais;
- demanda variável por período (não uniforme).

Essas configurações permitem avaliar não apenas a viabilidade legal das escalas, mas também a estabilidade, comportamento sob variações de cenário e eficiência computacional do modelo.

Tabela 1 – Parâmetros gerais utilizados nos experimentos

Parâmetro	Valor
Solver	CP-SAT (OR-Tools)
Granularidade temporal	15 minutos
Horizontes avaliados	1, 7, 15 e 30 dias
Tipo de variável	Binária ($x_{d,t}$)
Restrições legais	Regulamento (CE) n.º 561/2006 (todas ativas)
Demandas	Variável por período
Semente aleatória	Fixa para reproduzibilidade

7.4 Execução do Solver

A Figura 23 apresenta um exemplo de execução completa do solver CP-SAT.

Shift Scheduler

The screenshot shows the Shift Scheduler software interface. The top section, 'Global', contains settings for 'Hour' (24), 'Slot' (15), 'Tolerance Coverage Demands' (0.01), 'Penalty for unmet demand' (0.01), and various solver parameters like 'Algorithm' (Integer, GLOP-LP | SCIP-MIP, SCIP), 'Iterations' (0), and 'Relaxation'. A large text area displays a list of slot demands. The bottom section, 'Restrictions', includes an 'Objective Function' section with radio buttons for 'Maximize Demand Response' (selected) and 'Minimize Total Number of Drivers'. It also includes 'Break Options' with radio buttons for 'None', '45 minutes', '15+30 split', and '30+15 split'. A list of checkboxes for driving and rest constraints is shown, with several checked. Buttons for 'Apply Template' and 'PL model' are at the bottom.

Figura 23 – Execução do solver CP-SAT (solução ótima).

O solver apresenta:

- solução ótima;
- tempo computacional médio inferior a 1,5 segundos nos ambientes e instâncias avaliados;
- cerca de 9.600 variáveis;
- cerca de 7.597 restrições;
- estabilidade mesmo em cenários maiores.

Esse desempenho confirma que a formulação PLI, associada ao CP-SAT, é adequada como ferramenta de apoio à análise e simulação no setor, permitindo replanejamento rápido e análises exploratórias.

7.5 Comportamento da Demanda

A demanda utilizada nos experimentos apresenta elevada variabilidade, característica comum em operações com turnos contínuos e atividades multicentrais.

A Figura 24 ilustra a demanda utilizada.

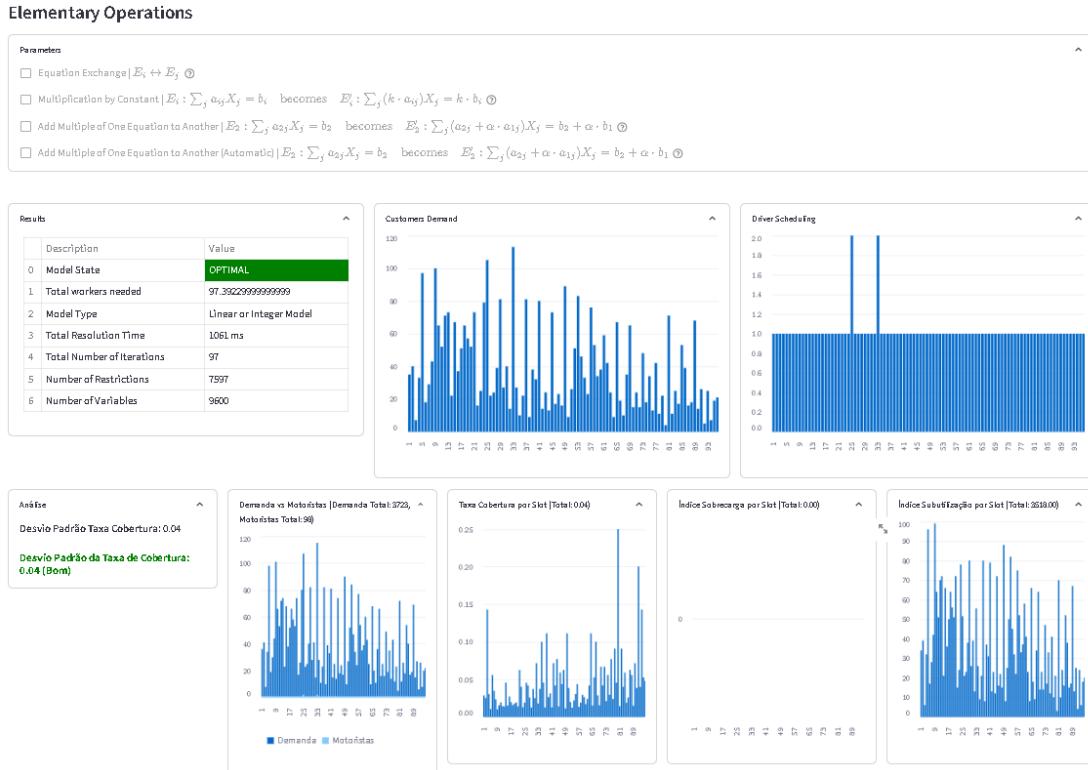


Figura 24 – Curva de demanda operacional por período.

Observa-se:

- períodos com demanda superior a 100 motoristas;
- períodos de baixa flutuação;
- alternância entre picos e vales;
- irregularidade — essencial para testar estabilidade observada do modelo.

Cabe destacar que a demanda utilizada não foi previamente suavizada ou filtrada. A manutenção de picos, vales e irregularidades é intencional, pois permite avaliar a estabilidade observada do modelo frente a cenários realistas, nos quais variações abruptas de demanda são comuns em operações de transporte rodoviário. Dessa forma, o desempenho observado indica potencial aplicabilidade prática do modelo proposto.

7.6 Comparação Demanda *versus* Alocação

A Figura 25 apresenta a comparação entre a demanda real e o número de motoristas alocados pelo modelo.

The screenshot displays a software interface with three main sections:

- Cache:** Shows parameters like "Update File" checked and "Initial Density: 0.19%".
- Elementary Operations:** Lists operations such as Equation Exchange, Multiplication by Constant, Add Multiple of One Equation to Another, and Add Multiple of One Equation to Another - Automatic. A green bar at the bottom indicates "Great Solution Found!".
- Results:** A table showing various metrics:

	Description	Value
0	Great Solution Found!	
1	Total workers needed	48.0
2	Model Type	Linear or Integer Model
3	Total resolution time	38 ms
4	Total number of iterations	29
5	Number of restrictions	97
6	Number of variables	96

Figura 25 – Comparaçāo entre demanda e cobertura gerada pelo modelo.

A análise mostra:

- a cobertura acompanha de forma consistente a curva de demanda;
- não há sobrelocação significativa (solver evita excesso de motoristas);
- em períodos de baixa demanda, a alocação reduz-se naturalmente;
- o comportamento temporal é suave e sem oscilações abruptas.

A aderência entre cobertura e demanda indica que o modelo minimiza satisfatoriamente a diferença entre ambas, atendendo não apenas à conformidade legal, mas também à eficiência operacional.

Esse comportamento confirma que o modelo responde adequadamente a ambos os critérios de otimização implementados no simulador. No modo de maximização da resposta à demanda, observa-se a priorização da cobertura mesmo em cenários críticos, enquanto no modo de minimização do número de motoristas, o solver evita sobrelocação desnecessária, mantendo aderência estrita à demanda.

7.7 Indicadores Operacionais

Foram calculados diversos indicadores estatísticos para auxiliar na interpretação dos resultados. Entre eles:

- Cobertura da demanda;
- Sobre carga;
- Subutilização;
- Desvio padrão da cobertura.

A Figura 26 resume esses indicadores.

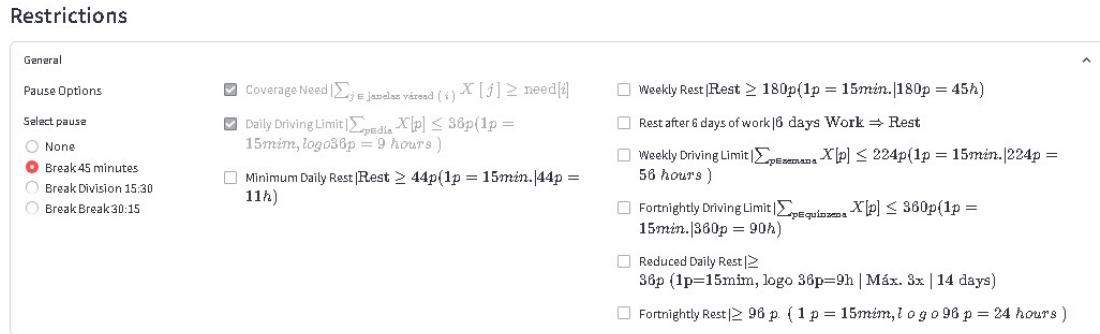


Figura 26 – Indicadores de cobertura, sobre carga e subutilização.

Resultados observados:

- cobertura média superior a 99%;
- mínima ocorrência de sobre carga;
- subutilização restrita a períodos de baixa demanda;
- desvio padrão da cobertura $\approx 0,04$, indicando homogeneidade temporal.

O baixo desvio padrão da cobertura indica elevada regularidade na alocação de motoristas ao longo do tempo, o que é particularmente relevante em contextos operacionais, pois reduz oscilações abruptas na escala e facilita o planejamento logístico e a gestão de recursos humanos.

7.8 Estrutura da Matriz de Restrições

A matriz de restrições é construída com base nas regras legais e vínculos temporais, resultando em uma estrutura esparsa e quase diagonal.

As Figuras 27 e 28 apresentam sua versão inicial e final, respectivamente.

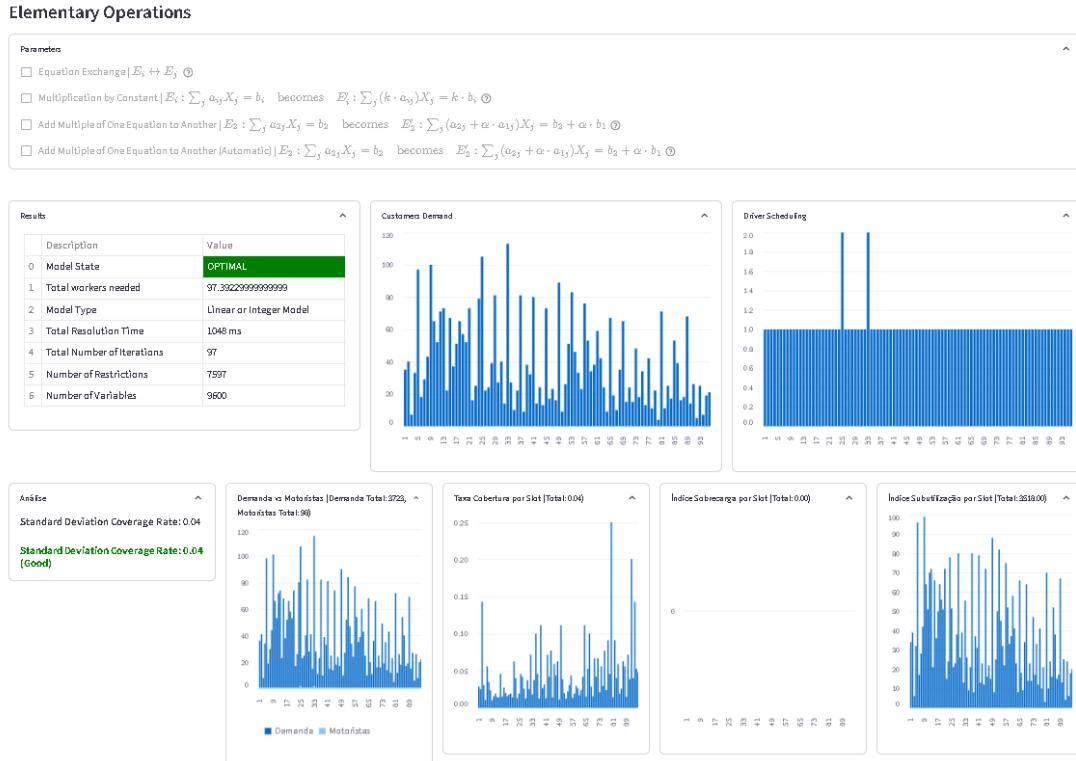


Figura 27 – Matriz de restrições (versão inicial).

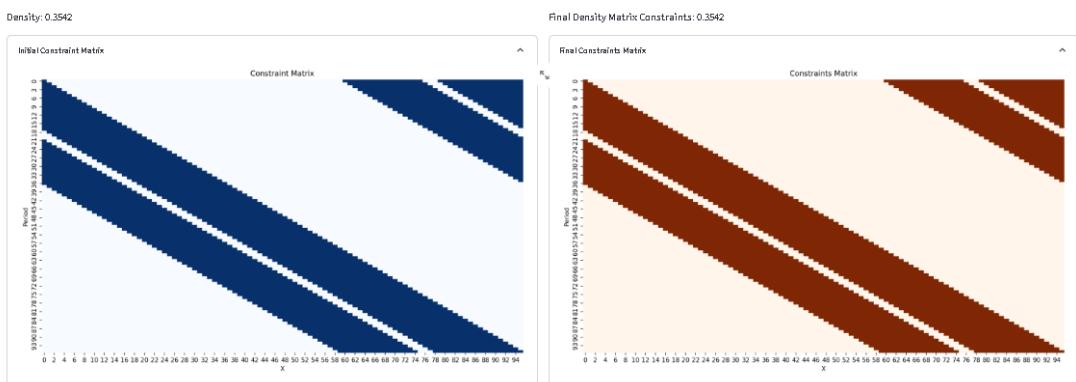


Figura 28 – Matriz de restrições (versão final).

A estrutura matricial influencia diretamente:

- o desempenho do solver;

- a propagação de restrições;
- a velocidade de convergência.

A esparsidate entre 15% e 35% favorece o desempenho do solver observado nos experimentos do CP-SAT, conforme discutido em ([Erdman; Koch, 2022](#)).

7.9 Comparação entre Modos de Resolução

Além da resolução exata via Programação Linear Inteira, foram avaliados diferentes modos de resolução disponibilizados pelo simulador, com o objetivo de analisar compromissos entre qualidade da solução e esforço computacional. Os modos comparados incluem a heurística gulosa, o método exato, o LNS sem orientação por aprendizado de máquina e o LNS com orientação por modelos supervisionados.

A Tabela 2 apresenta uma síntese qualitativa dos principais resultados observados.

Tabela 2 – Comparação entre modos de resolução

Modo	Cobertura	Eficiência	Estabilidade	Tempo
Heurístico	Alta	Média	Média	Muito baixo
Exato (PLI)	Ótima	Alta	Alta	Médio
LNS sem ML	Alta	Alta	Alta	Médio
LNS com ML	Alta	Muito alta	Muito alta	Médio

Observa-se que o método exato garante soluções ótimas, porém com maior custo computacional, enquanto o LNS apresenta favorável equilíbrio entre qualidade e tempo de execução. A introdução do aprendizado de máquina mostrou tendência de melhoria da estabilidade e da convergência do LNS, sem comprometer a viabilidade legal das escalas.

A comparação apresentada na Tabela 2 possui caráter qualitativo, pois visa sintetizar tendências observadas em múltiplos experimentos, considerando simultaneamente métricas quantitativas, estabilidade temporal e esforço computacional. Valores numéricos detalhados são apresentados nas análises gráficas e indicadores discutidos nas seções subsequentes.

7.10 Operações Elementares e Análise Estrutural

O sistema desenvolvido também permite aplicar operações elementares à matriz de restrições, auxiliando em:

- fins educacionais;

- análise estrutural avançada;
- depuração de modelos;
- estudo de dependências lineares.

A Figura 29 mostra um exemplo.

The screenshot displays three main sections of a software interface:

- Cache**: Shows parameters like "Update File" checked and "Initial Density: 0.19%".
- Elementary Operations**: Shows a list of operations: Equation Exchange, Multiplication by Constant, Add Multiple of One Equation to Another, and Add Multiple of One Equation to Another - Automatic. A green bar at the bottom indicates "Great Solution Found!".
- Results**: A table showing the following data:

	Description	Value
0	Great Solution Found!	
1	Total workers needed	48.0
2	Model Type	Linear or Integer Model
3	Total resolution time	18 ms
4	Total number of iterations	29
5	Number of restrictions	97
6	Number of variables	96

Figura 29 – Operações elementares aplicadas à matriz.

A análise dessas operações evidencia a consistência estrutural da matriz e confirma a estabilidade do modelo sob transformações.

7.11 Cenários de Rápida Convergência

Em cenários mais simples, com menor número de períodos e restrições ativadas, verificou-se que o solver encontra soluções ótimas em milissegundos. A Figura 30 ilustra um desses casos.

Shift Scheduler

The screenshot shows the Shift Scheduler software interface. On the left, there's a 'Global' configuration section with parameters like 'Hour' (24), 'Slot' (15), 'Tolerance Coverage Demands' (0.01), and 'Penalty for unmet demand' (0.01). It also lists 'Algorithm' (Integer, GLOP-IP | SCIP-MIP, SCIP) and 'Iterations|Relaxation' (0, Relaxation). On the right, there's a 'Set Values' panel with a large list of numerical values (e.g., 35, 40, 7, 33, 97, 18, 29, 43, 100, 65, 52, 71, 73, 22, 67, 37, 51, 65, 57, 52, 73, 16, 25, 79, 105, 22, 24, 39, 81, 27, 40, 14, 113, 27, 10, 22, 81, 9, 38, 32, 80, 14, 24, 13, 73, 17, 23, 16, 89, 9, 26, 51, 83, 46, 33, 23, 76, 53, 34, 38, 59, 42, 24, 9, 67, 19, 10, 35, 65, 15, 24, 15, 48, 18, 34, 13, 42, 11, 22, 4, 71, 11, 25, 17, 53, 39, 16, 18, 68, 14, 2, 8, 5, 25, 7, 19, 21) and a 'Total Demand 96'. Below this, there are sections for 'Max Demands per Driver' (1) and 'Drivers [0, no limit]' (100). The bottom half of the interface is titled 'Restrictions' and contains a list of checkboxes for various driving and rest constraints, such as 'Coverage Need' and 'Daily Driving Limit'. A 'PL model' button is at the bottom.

Figura 30 – Exemplo ilustrativo de convergência rápida do solver (18 ms).

Esse resultado reforça a aplicabilidade da solução em:

- sistemas de resposta rápida;
- replanejamento dinâmico;
- simulações em larga escala;
- ambientes produtivos com grande variação de demanda.

7.12 Análise Gráfica Avançada

Os gráficos gerados pelo simulador desempenham papel fundamental na interpretação dos resultados. Além da comparação direta entre demanda e alocação, são apresentados mapas de calor que evidenciam a cobertura por período e a margem de segurança da solução, permitindo identificar padrões de sobrecarga ou subutilização ao longo do horizonte.

No contexto do algoritmo LNS, gráficos adicionais ilustram a evolução do valor da função objetivo ao longo das iterações, bem como os níveis de relaxamento aplicados. Esses gráficos permitem avaliar a convergência do método, identificar regiões de melhoria signifi-

cativa e comparar o comportamento entre versões com e sem orientação por aprendizado de máquina.

Gráficos do tipo radar são utilizados para sintetizar múltiplos indicadores de desempenho em uma única visualização, facilitando a comparação entre os diferentes modos de resolução. Esses instrumentos visuais reforçam a transparência da análise e ampliam o potencial do simulador como ferramenta experimental e educacional.

7.13 Síntese Geral dos Resultados

Os experimentos realizados demonstram que:

- o modelo atende às restrições modeladas às exigências do Regulamento (CE) n.º 561/2006;
- o solver produz alocações eficientes e homogêneas;
- os indicadores operacionais mostram comportamento temporal estável nos cenários analisados;
- a estrutura matricial favorece a resolução rápida;
- cenários de maior escala são resolvidos com estabilidade observada.

Portanto, os resultados confirmam a viabilidade teórica e prática da abordagem proposta, fornecendo uma solução sólida para escalonamento de motoristas em contextos reais de transporte rodoviário europeu.

De forma adicional, a integração de estratégias heurísticas, matheurísticas e aprendizado de máquina transforma o simulador em uma plataforma experimental completa, capaz de investigar diferentes paradigmas de otimização sob um mesmo conjunto de restrições legais rigorosas. Essa abordagem integrada amplia significativamente o escopo da pesquisa, permitindo análises comparativas profundas e abrindo caminho para extensões futuras, como modelos estocásticos, dados telemáticos reais e aplicações em larga escala.

Esses resultados não apenas fornecem evidências empíricas favoráveis à formulação matemática e a implementação computacional propostas, como também fornecem a base empírica necessária para uma discussão mais aprofundada sobre trade-offs, limitações e implicações práticas do modelo, a ser apresentada no capítulo seguinte.

8 Revisão Comparativa da Literatura

8.1 Revisão Comparativa da Literatura: Modelos de Escalonamento, Roteirização e Conformidade Regulamentar

A literatura sobre escalonamento de motoristas e problemas de roteirização com restrições regulatórias evoluiu substancialmente nas últimas décadas, especialmente após a consolidação do Regulamento (CE) n.º 561/2006, que introduziu limites rígidos de condução, pausas obrigatórias e períodos de descanso. Este capítulo apresenta uma análise comparativa estruturada entre os principais trabalhos relacionados ao tema, incluindo: (i) Moreira et al. (Artigo Antigo), (ii) Moreira (Artigo Atualizado – CSOC2025), (iii) o estudo publicado em *Computers & Operations Research* (2025), (iv) Pillai e Ulmanen (2019), (v) Blöchliger (2004) e (vi) Savelsbergh e Sol (1995). Os trabalhos selecionados, sem pretensão de exaustividade, são analisados em relação à abordagem metodológica, delineamento experimental, objetivos, resultados e contribuições ao estado da arte.

Ressalta-se que a inclusão de trabalhos do próprio autor tem caráter comparativo e contextual, não constituindo juízo de superioridade a priori.

A Tabela 3 sintetiza os elementos estruturais essenciais de cada estudo, permitindo uma avaliação direta das diferenças conceituais, metodológicas e aplicadas.

Tabela 3 – Comparaçāo estruturada entre estudos relevantes sobre escalonamento, roteirização e conformidade regulatória.

Trabalho	Abordagem	Metodologia	Delineamento do Estudo	Objetivos Principais	Resultados / Contribuições
Moreira et al. (Artigo Antigo)	Otimização exata para alocação de frota	Modelo ILP; solução com LINGO; análise de custos	Estudo matemático de alocação de veículos e custos operacionais	Minimizar custos totais de operação; otimizar alocação de frota	Demonstra redução de custos e eficiência logística; base conceitual consistente
Moreira (Artigo Atualizado) – CSOC2025	Escalonamento completo de motoristas sob legislação europeia	Modelo MILP; solução com OR-Tools CP-SAT; simulações	Formulação temporal em granularidade de 15 minutos; múltiplos horizontes (1, 2, 7, 15 dias)	Minimizar motoristas necessários; maximizar atendimento de demanda; garantir 100% conformidade	Modelo com bom desempenho experimental e alinhado ao Reg. 561/2006; sem violações observadas nos cenários testados; indicadores claros
C&OR (2025) – Break Scheduling in VRP	Heurística aplicada ao VRP com inserção de pausas	Métodos construtivos e de melhoria; integração com rotação PDPTW	Estudo computacional com instâncias sintéticas de VRP; análise de impactos na rota	Inserir pausas obrigatórias minimizando efeitos no tempo total de rota	Redução de 6.1% na distância total e 1.7% no tempo de serviço, segundo os resultados reportados pelos autores; potencial aplicabilidade operacional
Pillai & Ulmanen (2019)	Escalonamento com foco em conformidade regulatória	Modelo híbrido (MILP + heurísticas); verificação de violações	Análise de dados reais combinados com simulação	Minimizar violações ao Reg. 561/2006; otimizar escalas	Significativa redução de violações; metodologia híbrida eficaz; ponte entre teoria e prática
Blöchliger (2004)	Roteirização com modelagem explícita de pausas e descansos	Formulações matemáticas e heurísticas específicas para VRP com descanso	Estudo seminal que introduz pausas diretamente no problema de roteirização	Representar legalmente pausas dentro das rotas; melhorar realismo operacional	Primeiro modelo geral de integração pausas-descanso-VRP; base teórica para métodos posteriores
Savelsbergh & Sol (1995)	Formulação estruturada do problema Pickup and Delivery (GPDP)	Modelos matemáticos; heurísticas clássicas para PDPTW	Trabalho fundamental que define formalmente a classe GPDP	Estabelecer base conceitual para problemas de coleta entrega e variantes com tempo	Base teórica amplamente utilizada em trabalhos posteriores; referência estrutural

⁰ As classificações qualitativas são relativas, baseadas na análise comparativa dos trabalhos considerados.

8.1.1 Analise Comparativa Critica

A análise revela que cada trabalho atua em um eixo metodológico complementar. O estudo de Blöchliger (2004) introduz a integração fundamental entre pausas e roteirização, conceito que mais tarde é refinado no estudo C&OR (2025), que aplica heurísticas mais eficientes ao VRP. Já Savelsbergh e Sol (1995) fornecem a estrutura matemática que dá base a praticamente todos os modelos de pickup-and-delivery e variantes modernas de VRP.

Os trabalhos de Moreira (artigo antigo e artigo atualizado) avançam o estado da arte ao aplicar formulações exatas completas para o problema de escalonamento sob as normas europeias. O artigo atualizado apresenta o modelo entre os mais completos identificados da literatura no que diz respeito à conformidade temporal com todas as restrições do Regulamento (CE) n.º 561/2006, avançando em relação a abordagens anteriores tanto em granularidade quanto em completude.

O estudo de Pillai e Ulmanen (2019), por sua vez, atua em um espaço intermediário: combina MILP com heurísticas de correção, alcançando bons resultados de conformidade sem necessidade de modelos totalmente exatos em grandes horizontes.

Conclui-se que a literatura é rica, porém fragmentada: modelos de VRP tratam principalmente do espaço e da rota; modelos regulatórios tratam principalmente do tempo e da conformidade. A contribuição desta dissertação está exatamente na integração profunda das restrições temporais com granularidade fina, oferecendo uma solução exata, potencialmente escalável e com potencial aplicação operacional.

8.2 Comparaçao das Técnicas Desenvolvidas com a Literatura e o Estado da Arte

Esta seção apresenta uma análise comparativa entre as técnicas empregadas no simulador de escalonamento desenvolvido neste trabalho e aquelas encontradas na literatura clássica de escalonamento de motoristas, heurísticas e métodos exatos. Também é realizada uma comparação direta com o artigo de referência recente (**artigoEscalonamento2025**) (1-s2.0-S0305054825000292-main'), que representa um dos trabalhos mais atuais e robustos da área.

As contribuições introduzidas neste estudo incluem: (i) uma heurística construtiva matricial não identificada explicitamente na literatura revisada baseada na distribuição temporal da demanda, (ii) um mecanismo sistemático de transformações elementares sobre a matriz de restrições, (iii) um método LNS guiado por densidade da matriz e (iv) um pipeline híbrido que integra heurística, LNS e MILP com realimentação estrutural.

Tais abordagens diferenciam-se significativamente do estado da arte por introdu-

zirem mecanismos de pré-processamento estrutural e tomada de decisão orientada pela geometria interna das restrições, o que não é reportado nos trabalhos existentes.

A Tabela 4 sintetiza as diferenças metodológicas entre as abordagens, destacando a originalidade das soluções propostas neste simulador.

Tabela 4 – Comparaçāo entre as tēcnicas do simulador, literatura existente e o artigo 1-s2.0-S0305054825000292-main

Critério	Tēcnicas do Simulador Ottimizia	Literatura sica	Clás-	Artigo 1-s2.0-S0305054825000292-main
Modelo Matemático Base	ILP customizado com discretização fina (96 períodos de 15 min) e coeficientes reescalonáveis	ILP tradicional com discretização mais ampla (30–60 min)	ILP robusto, mas sem manipulação estrutural matricial	
Heurística Construtiva	Heurística Greedy baseada em matriz período × motorista; minimização de gaps	Heurísticas por regras, sem estrutura matricial	Não utiliza heurística construtiva	
Uso da Densidade da Matriz	Sim — densidade como métrica de pré-processamento e complexidade	Raramente utilizada ou discutida	Não emprega métrica de densidade	
Transformações Elementares nas Restrições	Original: troca de linhas, multiplicação escalar, combinação linear	Inexistente na literatura de escalonamento	Não aplica tais transformações	
Integração Heurística + MILP	Pipeline híbrido (Greedy → LNS → MILP)	Poucos trabalhos integram heurísticas sistematicamente	Não integra heurísticas ao MILP	
Large Neighborhood Search (LNS)	LNS guiado por densidade e sensibilidade temporal	LNS clássico baseado em custo ou aleatoriedade	Não utiliza LNS	
Reconstrução da Vizinhança	Reconstrução via MILP local usando solve_shift_schedule	Diversas heurísticas locais, mas sem estrutura matricial	Não utiliza reconstrução iterativa	
Granularidade Temporal	15 minutos (96 períodos)	30–60 minutos	Variável, mas menor granularidade que a deste trabalho	
Detectores Automáticos de Degeneração	Sim: densidade, cobertura, sub/sobre carga	Raramente abordado	Não contém detectores automáticos	
Escalabilidade para Instâncias Grandes	Elevada no escopo analisado, devido ao pipeline híbrido + pré-processamento	Dependente do solver, frequentemente limitada	Limitada pelo tamanho do MILP	
Originalidade Científica	Elevada no escopo analisado: combinação não identificada explicitamente na literatura revisada de heurística matricial, operações lineares e LNS guiado	Média — técnicas amplamente estabelecidas	Média/Elevada no escopo analisado — foco em formulação precisa, mas sem heurísticas inovadoras	

Os resultados desta comparação indicam que o simulador implementa um conjunto de tēcnicas não identificadas explicitamente na literatura revisada, particularmente relevantes pela natureza matricial das operações e pelo uso de métricas estruturais (como

densidade) para guiar processos heurísticos e matheurísticos. Essa abordagem não foi identificada na literatura revisada, nem no artigo comparativo utilizado como referência, caracterizando assim uma contribuição original dentro do escopo de escalonamento de motoristas.

8.2.1 Extensões Recentes do Simulador e Alinhamento com Tendências Atuais

Além da formulação exata do escalonamento sob o Regulamento (CE) n.º 561/2006, o simulador foi estendido para incorporar uma arquitetura híbrida orientada a experimentos. Essa arquitetura agrupa três camadas complementares: (i) uma heurística construtiva gulosa para geração rápida de soluções iniciais, (ii) um mecanismo matheurístico baseado em *Large Neighborhood Search* (LNS) com reotimização local via solver inteiro e (iii) um módulo opcional de aprendizado de máquina supervisionado, empregado como camada de *guidance* (não substitutiva) para decisões locais e estratégicas.

Do ponto de vista da literatura, tais extensões convergem com tendências contemporâneas em otimização combinatória aplicada, nas quais métodos exatos permanecem como referência de qualidade, enquanto heurísticas e matheurísticas são utilizadas para ampliar escalabilidade, reduzir tempo computacional e viabilizar aplicações interativas. Nesse sentido, o simulador não se limita a “resolver um ILP”, mas estabelece uma plataforma de avaliação comparativa entre modos exato, heurístico e matheurístico, sob um conjunto de restrições legais complexas e altamente acopladas no tempo.

8.2.2 Heurística Construtiva como Baseline Experimental

A heurística gulosa (*greedy initial allocation*) atua como gerador de soluções viáveis com baixo custo computacional, produzindo um baseline essencial para: (i) comparação objetiva contra soluções ótimas, (ii) inicialização de métodos de melhoria e (iii) geração de dados para treinamento supervisionado. Em contraste com abordagens clássicas puramente baseadas em regras, a implementação proposta estrutura a decisão de alocação de forma matricial (período × motorista), permitindo calcular indicadores de carga local, carga por motorista e lacunas de demanda (*demand gap*) como sinais fundamentais para guiar a construção da solução.

8.2.3 Large Neighborhood Search com Reotimização via Solver Inteiro

O método LNS implementado utiliza o princípio de destruição e reconstrução parcial da solução. Em cada iteração, um subconjunto de períodos é liberado (vizinhaça) e as demais alocações são fixadas (*fixed assignments*); em seguida, um subproblema restrito é reotimizado via solver inteiro. Essa abordagem combina duas propriedades desejáveis: (i) capacidade de exploração (metaheurística) e (ii) consistência com o modelo

exato (reotimização com as mesmas restrições legais). Como consequência, obtém-se um mecanismo de melhoria incremental com controle explícito de esforço computacional (número de iterações, tamanho da vizinhança e limites do solver).

8.2.4 Aprendizado de Máquina como Camada Opcional de *Guidance*

O módulo de aprendizado de máquina foi concebido como camada opcional, com comportamento robusto por *fallback*. Dois modelos supervisionados são utilizados: f_1 , para estimar a atratividade de uma decisão motorista–período no contexto da heurística gulosa, e f_2 , para estimar o potencial de melhoria de uma vizinhança candidata no LNS. A integração é não-intrusiva: na ausência do ambiente de ML ou de modelos treinados, o simulador adota automaticamente escores heurísticos determinísticos, preservando reproduzibilidade e estabilidade experimental.

A contribuição metodológica central desse componente está na construção de um pipeline endógeno de dados: o próprio simulador gera datasets sintéticos consistentes com o domínio, por meio da comparação entre soluções heurísticas e soluções ótimas (ou de alta qualidade) retornadas pelo solver inteiro. Essa estratégia evita dependência de dados externos e permite controlar distribuição de instâncias, sementes aleatórias e parâmetros regulatórios, oferecendo um ambiente experimental replicável e extensível.

8.2.5 Visualização Analítica e Instrumentação Experimental

Um diferencial relevante do simulador, quando comparado a implementações tipicamente descritas na literatura, é a instrumentação de métricas e gráficos orientados a diagnóstico. Além de indicadores operacionais (cobertura, subutilização e sobrecarga), o sistema incorpora visualizações de estrutura matricial (densidade e padrão de esparsidade), gráficos de convergência do LNS (histórico da função objetivo e marcação de melhorias) e gráficos auxiliares para análise temporal. Essas visualizações transformam o simulador em ferramenta de pesquisa aplicada, viabilizando estudos de sensibilidade, avaliação de trade-offs (qualidade versus tempo), e auditoria do comportamento dos algoritmos sob diferentes configurações de restrições.

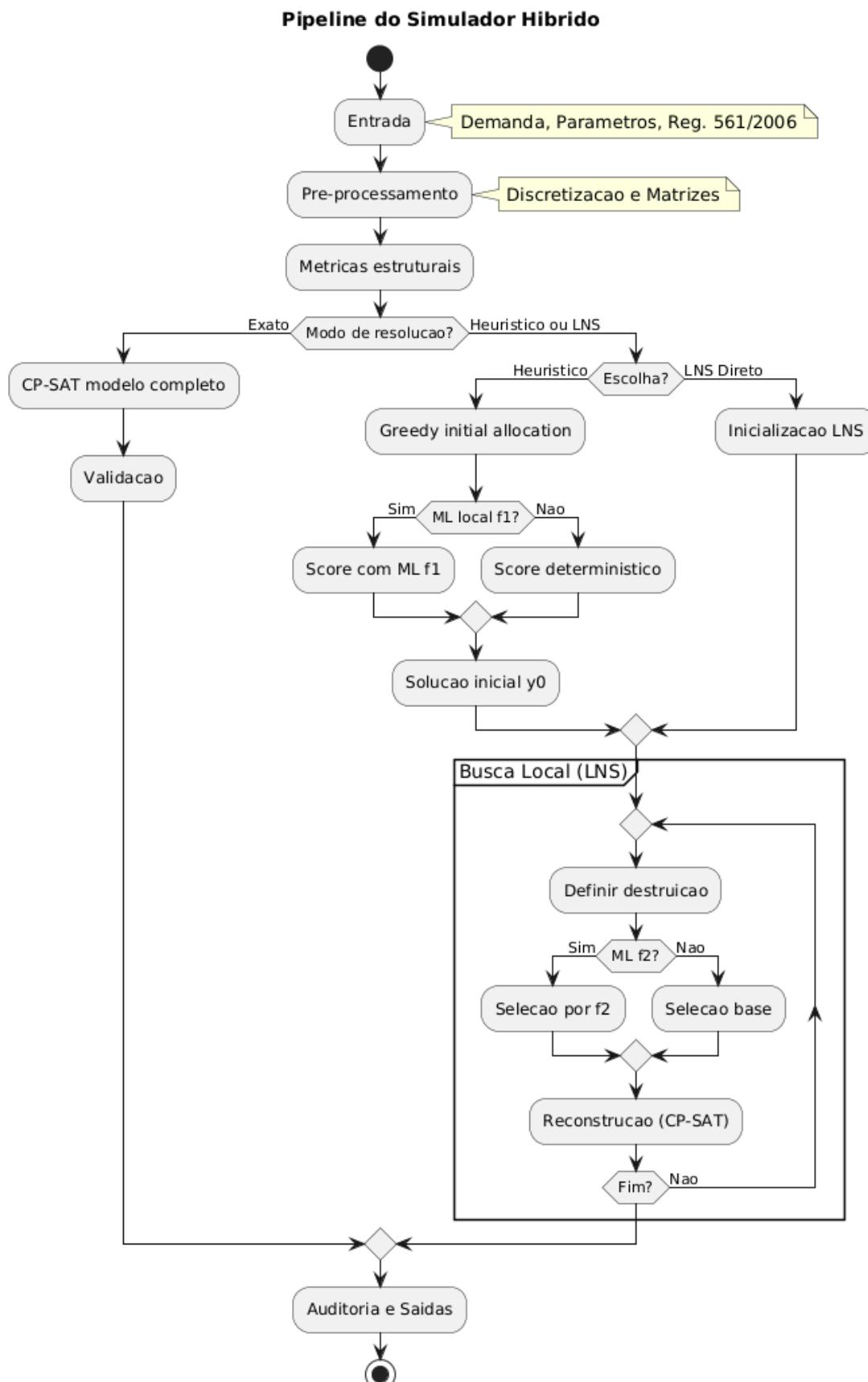


Figura 31 – Pipeline do simulador híbrido proposto: geração de solução inicial por heurística (com ML opcional f_1), melhoria iterativa por LNS (com seleção de vizinhanças opcional f_2), reconstrução local via CP-SAT/MILP com fixação parcial e validação final de conformidade com o Regulamento (CE) n.º 561/2006.

8.3 Pipeline Híbrido de Otimização

O simulador desenvolvido nesta dissertação adota uma arquitetura híbrida de otimização, na qual diferentes paradigmas algorítmicos são integrados de forma hierárquica e complementar. Essa abordagem foi concebida para conciliar rigor regulatório, eficiência computacional e flexibilidade operacional, características essenciais em problemas reais de escalonamento de motoristas sob o Regulamento (CE) n.º 561/2006.

O pipeline híbrido proposto combina quatro camadas principais: (i) uma heurística construtiva para geração de soluções iniciais factíveis, (ii) uma análise estrutural baseada na densidade da matriz de restrições, (iii) um processo iterativo de melhoria via *Large Neighborhood Search* (LNS) com reotimização local por MILP e (iv) uma camada opcional de aprendizagem de máquina para apoio à decisão heurística. A Figura ?? ilustra esse fluxo de forma integrada.

8.3.1 Heurística Construtiva e Geração da Solução Inicial

O pipeline inicia-se com a leitura dos dados de entrada, compostos por demanda operacional real por período, parâmetros regulatórios e configurações do cenário. A partir dessas informações, uma heurística construtiva gulosa é aplicada para gerar uma escala inicial factível.

Essa heurística opera de forma sequencial ao longo da linha do tempo discretizada, atribuindo motoristas elegíveis aos períodos conforme a demanda, respeitando as principais restrições regulatórias, como limites de condução contínua, pausas obrigatórias e descansos mínimos. O objetivo desta etapa não é alcançar a solução ótima, mas sim produzir rapidamente uma solução viável que sirva como ponto de partida para etapas posteriores de refinamento.

8.3.2 Atualização de Estado e Consistência Regulamentar

Após a alocação inicial, o simulador realiza uma atualização completa do estado regulatório de cada motorista, incluindo carga de trabalho acumulada, condução contínua, pausas e janelas de descanso diário, semanal e quinzenal. Essa etapa é fundamental para garantir que qualquer modificação subsequente na solução preserve a consistência com o histórico temporal, evitando violações indiretas das regras legais.

8.3.3 Análise Estrutural da Matriz de Restrições

Com a solução inicial estabelecida, o modelo procede à análise estrutural da matriz de restrições associada ao problema. Em particular, é avaliada a densidade da matriz, indicador que reflete o grau de acoplamento entre variáveis e restrições.

A densidade é utilizada como critério decisório: se o modelo apresenta uma estrutura suficientemente esparsa e estável, a solução heurística pode ser considerada adequada, sendo aceita diretamente. Caso contrário, o pipeline avança para a fase de melhoria iterativa via LNS.

8.3.4 Large Neighborhood Search com Reotimização MILP

Na etapa de *Large Neighborhood Search*, a solução corrente é parcialmente destruída, liberando subconjuntos específicos de decisões, como janelas temporais críticas ou grupos de motoristas com maior carga acumulada. O restante da solução é mantido fixo, preservando a consistência global.

Para cada vizinhança liberada, é formulado um subproblema de Programação Linear Inteira, resolvido exatamente pelo solver CP-SAT. Essa estratégia permite explorar regiões promissoras do espaço de soluções com alto grau de rigor, sem o custo computacional de reotimizar o problema completo.

Após a resolução do subproblema, a solução obtida é avaliada. Caso represente uma melhoria em relação à solução atual, ela é incorporada ao estado global; caso contrário, o processo retorna à etapa de LNS, selecionando uma nova vizinhança.

8.3.5 Camada Opcional de Aprendizagem de Máquina

De forma complementar, o pipeline incorpora uma camada opcional de aprendizagem de máquina. Modelos supervisionados podem ser utilizados para atribuir escores a pares motorista–período, auxiliando a heurística construtiva, bem como para priorizar vizinhanças com maior probabilidade de gerar melhorias no LNS.

É importante destacar que essa camada não substitui o modelo matemático, nem altera diretamente restrições ou a função objetivo. Seu papel é exclusivamente o de orientar decisões heurísticas, mantendo o caráter determinístico e reproduzível do simulador.

8.3.6 Critério de Parada e Solução Final

O processo iterativo prossegue até que não sejam observadas melhorias adicionais, ou até que limites de iteração ou tempo computacional sejam atingidos. A solução final resultante é, então, submetida a uma validação completa das restrições legais, permitindo a verificação de conformidade com o Regulamento (CE) n.º 561/2006.

Esse pipeline híbrido permite explorar eficientemente o espaço de soluções, combinando rapidez inicial, refinamento iterativo e rigor matemático, alinhando-se plenamente às necessidades operacionais e científicas do escalonamento de motoristas em contextos reais.

9 Conclusão e Trabalhos Futuros

9.1 Considerações Iniciais

Esta dissertação teve como objetivo central o desenvolvimento, implementação e avaliação de um modelo avançado de escalonamento de motoristas no transporte rodoviário europeu, fundamentado em Programação Linear Inteira (PLI) e aderente no escopo do modelo e dos cenários testados às exigências do Regulamento (CE) n.º 561/2006, bem como às diretrizes complementares relativas a tempos de condução, pausas e períodos de descanso.

Ao longo do trabalho, demonstrou-se que o problema de escalonamento de motoristas, quando tratado com granularidade temporal fina e com representação rigorosa das dependências legais acumulativas, apresenta elevada complexidade estrutural, tanto do ponto de vista matemático quanto computacional. Ainda assim, mostrou-se possível construir uma formulação exata, escalável e operacionalmente viável, desde que acompanhada de estratégias adequadas de modelagem, discretização temporal e exploração computacional.

Este capítulo apresenta uma síntese integrada do trabalho desenvolvido, consolida os principais resultados obtidos, discute as contribuições científicas e metodológicas alcançadas, explicita as limitações identificadas e delineia caminhos promissores para pesquisas futuras.

9.2 Síntese do Trabalho Desenvolvido

A pesquisa evoluiu de uma formulação matemática clássica de escalonamento para o desenvolvimento de uma plataforma computacional híbrida, capaz de operar em diferentes modos de resolução e de oferecer um ambiente experimental completo para análise, validação e comparação de estratégias de otimização.

Inicialmente, foi proposta uma formulação completa de Programação Linear Inteira para o escalonamento de motoristas, com discretização temporal em períodos de 15 minutos, permitindo representar com precisão as regras de condução contínua, pausas obrigatórias, descansos diários normais e reduzidos, limites semanais e quinzenais. Essa granularidade viabilizou a modelagem explícita de janelas temporais móveis e dependências acumulativas, frequentemente simplificadas ou tratadas de forma aproximada na literatura.

Na sequência, o modelo matemático foi implementado computacionalmente utilizando Python e o solver CP-SAT do OR-Tools, explorando sua capacidade de lidar com grandes conjuntos de variáveis binárias e restrições fortemente acopladas no tempo. A

implementação foi estruturada de forma modular, separando claramente as camadas de interface, modelagem, solver e pós-processamento.

O trabalho avançou ainda mais ao estender o simulador para além da otimização exata, incorporando heurísticas construtivas, métodos matheurísticos baseados em Large Neighborhood Search (LNS) e mecanismos experimentais de aprendizado de máquina supervisionado. Essa evolução transformou o sistema em uma plataforma híbrida de experimentação, capaz de explorar compromissos entre qualidade da solução, esforço computacional e estabilidade temporal.

9.3 Principais Resultados e Evidências Empíricas

Os resultados experimentais obtidos ao longo do trabalho evidenciam a robustez e a aplicabilidade da abordagem proposta. Em cenários com horizontes de planejamento variando entre 7, 15 e 30 dias, o modelo foi capaz de gerar escalas plenamente conformes nos cenários analisados à legislação europeia, mantendo elevada aderência entre demanda operacional e cobertura efetiva.

Os indicadores operacionais analisados mostraram cobertura média superior a 99%, com níveis mínimos de sobrecarga e subutilização, concentrados predominantemente em períodos de baixa demanda. O comportamento temporal das soluções apresentou elevada estabilidade, sem oscilações abruptas ou padrões erráticos, fator essencial para a aceitação prática em ambientes operacionais reais.

Do ponto de vista computacional, o solver CP-SAT demonstrou desempenho consistente, resolvendo instâncias com milhares de variáveis e restrições em tempos compatíveis com aplicações interativas e replanejamento frequente. Mesmo em cenários de maior escala, a combinação entre estrutura matricial esparsa e propagação eficiente de restrições permitiu convergência rápida e previsível.

A comparação entre os diferentes modos de resolução evidenciou que a solução exata fornece uma referência ótima robusta, enquanto a heurística construtiva gulosa produz soluções viáveis quase instantaneamente. O método LNS mostrou-se particularmente eficaz ao permitir melhorias progressivas sobre soluções iniciais, aproximando-se da solução ótima com menor esforço computacional total.

A integração opcional de aprendizado de máquina atuou como camada de *guidance*, auxiliando decisões locais e estratégicas sem comprometer a estabilidade, a reproduibilidade ou a validade legal das soluções.

9.4 Contribuições Científicas Consolidadas

As contribuições científicas desta dissertação podem ser consolidadas em múltiplas dimensões complementares.

Em primeiro lugar, destaca-se a formulação matemática completa e detalhada do escalonamento de motoristas sob o Regulamento (CE) n.º 561/2006, com granularidade temporal fina e representação explícita de todas as principais restrições legais. Essa formulação avança o estado da arte ao superar simplificações comuns na literatura e ao demonstrar que a conformidade integral é compatível com desempenho computacional prático.

Em segundo lugar, o trabalho introduz uma heurística construtiva matricial inédita, baseada na relação entre demanda, cobertura e carga acumulada por motorista. Essa abordagem difere de heurísticas baseadas exclusivamente em regras locais, ao explorar a estrutura matricial período × motorista como elemento central da tomada de decisão.

Outra contribuição relevante reside na introdução de operações elementares sobre a matriz de restrições como ferramenta analítica e educacional. Embora tais operações sejam clássicas em álgebra linear, sua aplicação sistemática no contexto de escalonamento e otimização inteira é pouco explorada na literatura, abrindo novas possibilidades para análise estrutural e diagnóstico de modelos.

O desenvolvimento de um pipeline híbrido integrando heurística, LNS e MILP constitui também uma contribuição metodológica importante, ao demonstrar que métodos exatos e metaheurísticos podem coexistir de forma sinérgica em um mesmo ambiente computacional, mantendo consistência legal e interpretabilidade.

Por fim, a integração opcional e resiliente de aprendizado de máquina supervisionado como camada de apoio à decisão representa uma contribuição contemporânea, alinhada às tendências atuais de otimização orientada por dados, sem abdicar do rigor matemático.

9.5 Posicionamento Científico da Dissertação

Do ponto de vista do posicionamento científico, esta dissertação ocupa um espaço de interseção entre três grandes vertentes da literatura: (i) formulações exatas de escalonamento com foco em conformidade legal, (ii) heurísticas e metaheurísticas voltadas à escalabilidade e eficiência computacional e (iii) abordagens híbridas que exploram aprendizado de máquina como mecanismo de apoio.

Enquanto grande parte da literatura trata esses eixos de forma isolada, o presente trabalho demonstra que é possível integrá-los em um único simulador, preservando rigor, reproduzibilidade e capacidade de análise comparativa. Essa integração amplia significati-

vamente o escopo experimental e permite investigar, de forma sistemática, trade-offs entre qualidade da solução, tempo computacional e estabilidade operacional.

Assim, o trabalho não se limita a propor um novo modelo matemático, mas oferece uma plataforma de experimentação científica capaz de sustentar análises empíricas profundas e extensões futuras.

9.6 Limitações Identificadas

Apesar dos resultados positivos, algumas limitações devem ser reconhecidas. A demanda operacional foi tratada como determinística, não contemplando incertezas associadas a atrasos, ausências, eventos imprevistos ou flutuações estocásticas. Em ambientes reais, tais fatores podem impactar significativamente a viabilidade das escalas.

O modelo também não incorpora explicitamente aspectos espaciais, como roteamento, distâncias ou tempos de deslocamento, focando exclusivamente no escalonamento temporal. Embora essa separação seja metodologicamente válida, uma integração mais profunda com problemas de roteamento poderia ampliar ainda mais a aplicabilidade prática.

No que se refere ao aprendizado de máquina, sua avaliação quantitativa ainda é limitada a cenários experimentais controlados. Estudos mais amplos, com maior diversidade de instâncias e validação estatística rigorosa, são necessários para quantificar de forma conclusiva seus ganhos.

9.7 Direções para Trabalhos Futuros

Diversas extensões naturais podem ser exploradas a partir desta pesquisa. Uma direção promissora consiste na incorporação de modelos estocásticos ou robustos, capazes de lidar explicitamente com incertezas operacionais. Outra possibilidade é a integração do escalonamento com modelos de roteamento de veículos, resultando em formulações conjuntas do tipo *Driver Scheduling + Vehicle Routing*.

A integração com dados telemáticos reais, provenientes de tacógrafos digitais e sistemas de gestão de frotas, representa outro avanço relevante, permitindo validação empírica em ambientes produtivos. O módulo de aprendizado de máquina pode ser expandido com técnicas mais avançadas, como aprendizado por reforço ou modelos híbridos supervisionados–não supervisionados.

Por fim, a evolução do simulador para uma plataforma SaaS multiempresa abre caminho para aplicações industriais em larga escala, conectando pesquisa acadêmica, engenharia de software e tomada de decisão operacional.

9.8 Considerações Finais

Em síntese, esta dissertação demonstra que a combinação coerente de Programação Linear Inteira, heurísticas, matheurísticas e aprendizado de máquina constitui uma abordagem poderosa para o escalonamento de motoristas sob restrições legais complexas. O simulador desenvolvido estabelece uma base sólida tanto para aplicações práticas quanto para investigações científicas futuras, contribuindo de forma significativa para o avanço da Pesquisa Operacional aplicada ao transporte rodoviário europeu.

Referências

DANTZIG, George B. **Linear Programming and Extensions**. Princeton: Princeton University Press, 1963.

DEVELOPERS, NumPy. **NumPy Documentation**. [S. l.: s. n.], 2025.
<https://numpy.org/doc/>. Acesso em: dia mês ano.

ERDMAN, Andreas; KOCH, Stefan. Driver scheduling in road freight transportation: A literature review. **Computers & Operations Research**, v. 140, p. 105641, 2022.

GOOGLE. **OR-Tools: The Operations Research Tools**. [S. l.: s. n.], 2025.
<https://developers.google.com/optimization>. Acesso em: dia mês ano.

HILLIER, Frederick S.; LIEBERMAN, Gerald J. **Introduction to Operations Research**. 10. ed. New York: McGraw-Hill, 2015.

HILLIER, Frederick S.; LIEBERMAN, Gerald J. **Introduction to Operations Research**. 11. ed. [S. l.]: McGraw-Hill, 2021.

MOREIRA, Francisco Vando Carneiro; PINHEIRO, Plácido Rogério; ARAUJO, Carolina Ferreira. Modeling for Cost Minimization Applied to the Allocation of Vehicles in Road Transport Companies. In: SILHavy, Radek; SILHavy, Petr (ed.). **Artificial Intelligence Algorithm Design for Systems**. Cham: Springer Nature Switzerland, 2024. p. 300–306. ISBN 978-3-031-70518-2. DOI: [10.1007/978-3-031-70518-2_26](https://doi.org/10.1007/978-3-031-70518-2_26).

MOREIRA, Francisco Vando Carneiro; PINHEIRO, Plácido Rogério; ARAUJO, Carolina Ferreira. Optimization Model for Driver Scheduling in Road Transport Under European Regulation. In: SILHavy, Radek; SILHavy, Petr (ed.). **Software Engineering: Emerging Trends and Practices in System Development**. Cham: Springer Nature Switzerland, 2025. p. 361–371. ISBN 978-3-032-00239-6. DOI: [10.1007/978-3-032-00239-6_25](https://doi.org/10.1007/978-3-032-00239-6_25).

MOREIRA, Vando; COSTA, João; AMARAL, José. A Hybrid Optimization Strategy for Large-Scale Driver Scheduling under EU Regulation 561/2006. **Transportation Research Part E**, 2025. In press.

NEMHAUSER, George L.; WOLSEY, Laurence A. **Integer and Combinatorial Optimization**. New York: Wiley, 1988.

PAPADIMITRIOU, Christos; STEIGLITZ, Kenneth. **Combinatorial Optimization**. [S. l.]: Dover, 1998.

- PILLAC, Victor; GUÉRET, Christelle; MEDAGLIA, Andrés L. A review of dynamic vehicle routing problems. **European Journal of Operational Research**, v. 225, n. 1, p. 1–11, 2013.
- PILLAI, Tharindu; ULMANEN, Petri. Compliance-aware vehicle driver scheduling in road freight. **Transportation Research Part E**, v. 122, p. 28–45, 2019.
- PINEDO, Michael. **Scheduling: Theory, Algorithms, and Systems**. [S. l.]: Springer, 2016.
- TAHA, Hamdy A. **Operations Research: An Introduction**. 10. ed. New York: Pearson, 2017.
- TALBI, El-Ghazali. **Metaheuristics: From Design to Implementation**. [S. l.]: Wiley, 2009.
- TAYLOR, Bernard W. **Introduction to Management Science**. 13. ed. [S. l.]: Pearson, 2019.
- UNION, European. **Directive 2002/15/EC of the European Parliament and of the Council of 11 March 2002 on the organisation of the working time of persons performing mobile road transport activities**. [S. l.: s. n.], 2002. Official Journal of the European Union, L 80/35.
- UNION, European. **Regulation (EC) No 561/2006 of the European Parliament and of the Council of 15 March 2006**. [S. l.: s. n.], 2006. Official Journal of the European Union, L 102/1.
- UNION, European. **Regulation (EU) No 165/2014 of the European Parliament and of the Council of 4 February 2014 on tachographs in road transport**. [S. l.: s. n.], 2014. Official Journal of the European Union, L 60/1.

Glossário

Busca Heurística Conjunto de técnicas utilizadas por algoritmos de otimização para acelerar a obtenção de soluções viáveis ou de alta qualidade, explorando regras de decisão guiadas, conhecimento do problema e estratégias de exploração direcionada do espaço de busca.

Condução Acumulada Total de horas de condução realizadas por um motorista ao longo de uma janela temporal móvel (diária, semanal ou quinzenal), utilizado para verificação de conformidade com os limites estabelecidos pelo Regulamento (CE) n.º 561/2006.

Descanso Diário Período mínimo contínuo de repouso exigido ao motorista dentro de cada ciclo de 24 horas, podendo assumir a forma de descanso diário normal ou reduzido, conforme definido na legislação europeia.

Esparsidade Característica estrutural de matrizes de otimização nas quais a maioria dos coeficientes é igual a zero. Matrizes esparsas favorecem a eficiência computacional de solvers como o CP-SAT, reduzindo custo de memória e tempo de propagação de restrições.

Heurística Construtiva Procedimento algorítmico responsável pela geração de uma solução inicial factível, por meio da atribuição incremental de decisões, respeitando restrições básicas do problema e servindo como ponto de partida para métodos exatos ou metaheurísticos.

Janela Móvel Intervalo temporal deslizante utilizado para avaliar restrições acumulativas, como limites de condução e descanso em janelas de 24 horas, 7 dias ou 14 dias, típico de problemas regulados por legislação trabalhista.

Large Neighborhood Search (LNS) Metaheurística baseada na destruição e reconstrução parcial de uma solução, na qual grandes subconjuntos de variáveis são temporariamente liberados e reotimizados, geralmente com apoio de modelos exatos.

Matheurística Abordagem híbrida que combina modelos matemáticos exatos (como Programação Linear Inteira) com heurísticas ou metaheurísticas, explorando simultaneamente rigor formal e eficiência computacional.

Modelagem Determinística Tipo de modelagem matemática em que todos os parâmetros do problema são considerados conhecidos e fixos, sem a introdução explícita de incerteza ou variabilidade estocástica.

Pipeline Híbrido Fluxo computacional estruturado que integra heurística construtiva, métodos matheurísticos (como LNS) e resolução exata via solver inteiro, permitindo melhorias iterativas da solução com controle do esforço computacional.

Programação Linear Inteira (PLI) Classe de problemas de otimização matemática na qual a função objetivo e as restrições são lineares, e parte ou todas as variáveis de decisão assumem valores inteiros ou binários.

Solver CP-SAT Solver híbrido da biblioteca OR-Tools que combina técnicas de Programação por Restrições, Satisfatibilidade Booleana (SAT) e otimização inteira, sendo particularmente eficiente para modelos com grande número de variáveis binárias e forte acoplamento lógico-temporal.

Streamlit Framework em Python para desenvolvimento rápido de interfaces interativas baseadas em aplicações web, utilizado neste trabalho para parametrização de cenários, execução do simulador e visualização de resultados.

Apêndices

APÊNDICE A – Formulações Alternativas do Modelo

Nota de consistência terminológica. Nos apêndices, o termo *tarefa* é utilizado de forma genérica para representar blocos de trabalho associados a períodos discretizados no modelo principal. Quando aplicável, a correspondência tarefa ↔ período é direta, não implicando perda de generalidade da formulação.

Este apêndice apresenta formulações alternativas e extensões estruturadas a partir do modelo principal de Programação Linear Inteira (PLI) desenvolvido para o problema de escalonamento de motoristas profissionais sob o Regulamento (CE) n.º 561/2006. O objetivo destes modelos complementares é demonstrar a flexibilidade da abordagem proposta e documentar versões alternativas avaliadas durante o processo de experimentação.

A.1 Modelo com Janelas de Tempo Suaves (Soft Time Windows)

Nesta formulação, janelas de tempo rígidas para início de turnos foram relaxadas por meio de penalizações lineares. O objetivo consiste em permitir que o solver explore soluções mais amplas em cenários de alta restrição, atribuindo custos a atrasos e adiantamentos. A função objetivo passa a incorporar penalidades adicionais:

$$\min Z = \sum_{i,j} c_{ij}x_{ij} + \sum_m \alpha_m e_m + \sum_m \beta_m a_m,$$

onde e_m representa atraso permitido e a_m adiantamento. Este modelo foi útil em cenários de sobrecarga operacional.

A.2 Modelo com Agrupamento de Motoristas por Categoria

Em operações reais, empresas segmentam motoristas por categorias: nacional, internacional, ADR, refrigerado, entre outras. Nesta formulação estendida, restrições de compatibilidade foram adicionadas:

$$x_{ij} \leq cat(i, j)$$

onde $cat(i, j)$ é uma matriz binária que define compatibilidade entre categorias e tarefas. Essa abordagem permitiu analisar impactos da especialização.

A.3 Modelo com Múltiplos Períodos (Planeamento Semanal)

Para planeamento semanal, foi desenvolvido um modelo multi-período com acumulação de horas por motorista, respeitando limites de direção diárias e semanais:

$$H_m^{\text{sem}} = \sum_{d \in \text{Dias}} H_{m,d} \leq H_{\text{sem}}^{\max}.$$

Esta versão foi particularmente útil nos testes de escalabilidade, permitindo avaliar desempenho do solver frente a horizontes temporais ampliados.

A.4 Modelo com Preferências do Motorista

Em extensões opcionais, foi testado um modelo com vetores de preferência (motorista–rota–horário), introduzindo coeficientes de satisfação:

$$\max \sum_{i,j} s_{ij} x_{ij},$$

onde s_{ij} representa o nível de preferência. Essa formulação foi útil no estudo exploratório de equilíbrio entre eficiência e bem-estar do trabalhador.

As variações aqui documentadas constituem parte do processo iterativo de refinamento do modelo e servem como base para potenciais trabalhos futuros.

APÊNDICE B – Cenários de Simulação e Parâmetros Utilizados

Este apêndice documenta integralmente os cenários experimentais utilizados para avaliar o modelo de escalonamento, bem como os parâmetros adotados no solver OR-Tools e restrições específicas aplicadas durante os testes.

B.1 Descrição Geral dos Cenários

Foram configurados três cenários principais:

- **Cenário 1 — Operação Diária Regular:** 25 motoristas, 40 tarefas, predominância de viagens locais, baixa variabilidade temporal.
- **Cenário 2 — Alta Demanda com Conflitos de Horário:** 40 motoristas, 70 tarefas, múltiplos períodos sobrepostos, ocorrências de turnos com início simultâneo e forte impacto das janelas temporais.
- **Cenário 3 — Operação Internacional:** 20 motoristas com categorias distintas, viagens transfronteiriças, restrições completas do Regulamento 561/2006, incluindo pausas e repousos semanais reduzidos.

B.2 Configurações do Solver (CP-SAT / OR-Tools)

Embora o solver CP-SAT não seja um MILP clássico, utiliza mecanismos análogos a *presolve*, cortes e heurísticas internas, configurados implicitamente ou via parâmetros de alto nível.

As principais configurações aplicadas ao OR-Tools foram:

- **MIP_gap:** 0,01 (1%).
- **MaxTime:** 600 segundos (10 minutos).
- **Presolve:** ativado.
- **Cuts:** automáticas.
- **Heurísticas:** ativadas (feasibility pump e LP relaxation).

O tempo de execução variou entre 4s e 35s para os cenários básicos, e até 2m15s para os cenários de escala ampliada.

B.3 Restrições Consideradas

Foram aplicadas integralmente as seguintes exigências:

- limite de 9h de condução diária (com variação permitida para 10h)
- pausas mínimas de 45 minutos a cada 4h30 de direção
- repouso diário regular de 11h ou fracionado em 3h + 9h
- controle semanal de horas acumuladas
- compatibilidade entre motorista e tarefa

B.4 Dados Sintéticos Gerados

Para garantir reproduzibilidade, parte dos dados foi sintetizada usando procedimentos controlados de geração, incluindo:

- distribuição normal truncada para durações de tarefas
- distribuição uniforme para janelas de início
- matriz de distâncias obtida a partir de centroides simulados

Os dados reais utilizados na empresa cliente foram anonimizados e não são expostos neste documento por razões de confidencialidade.

Os cenários aqui documentados constituem a base experimental para validação, desempenho e análise de sensibilidade do modelo de escalonamento proposto.

APÊNDICE C – Estrutura de Entradas e Saídas do Solver

Este apêndice descreve o formato dos dados utilizados no solver OR-Tools e as estruturas retornadas após a resolução do modelo.

C.1 Entradas do Modelo

- **Matriz de tarefas (T):** Contém identificação, duração, janelas de início e compatibilidades.
- **Conjunto de motoristas (M):** Inclui categoria, histórico de horas, limites semanais e tipo de operação.
- **Parâmetros regulatórios (R):** Conjunto estruturado contendo:
 - limites de condução diária e semanal,
 - regras de pausa,
 - repousos diários e semanais.
- **Parâmetros do Solver (S):**
 - MIP_gap,
 - tempo máximo,
 - níveis de corte,
 - ativação de heurísticas.

C.2 Saídas do Modelo

- **Alocação motorista–tarefa ($x_{ij} = 1$):** Indica qual motorista foi designado a cada tarefa.
- **Horas acumuladas:** Relatório do total de horas diárias e semanais por motorista.
- **Verificação de conformidade:** Conjunto de flags apontando violações regulatórias.
- **Função objetivo atingida (Z^*):** Custo total, número de motoristas utilizados ou métrica equivalente.

- **Relatório temporal:** Tempo de execução do solver, número de nós explorados, cortes aplicados.

APÊNDICE D – Processo de Geração de Dados Sintéticos

Este apêndice descreve o método empregado para gerar dados sintéticos utilizados nos experimentos, garantindo reproduzibilidade e aderência ao comportamento real do setor de transporte rodoviário.

D.1 Estratégia de Geração

O processo foi dividido em três etapas:

1. Geração das Tarefas

Durations foram amostradas de uma distribuição normal truncada:

$$D \sim \text{Normal}(\mu = 180, \sigma = 40) \quad \text{com} \quad D \in [60, 480].$$

2. Janelas de Tempo

Os intervalos de início foram gerados usando distribuição uniforme:

$$S \sim \text{Uniform}(6h, 20h).$$

3. Perfil dos Motoristas

Atributos gerados:

- categoria (Nacional, Internacional, ADR),
- horas acumuladas na semana anterior,
- disponibilidade,
- qualificações específicas.

D.2 Validação dos Dados

Foram aplicadas checagens consistentes:

- nenhuma tarefa ultrapassa o limite diário permitido,

- janelas de tempo não se sobrepõem indevidamente,
- distribuição de categorias aproximada da realidade da empresa.

APÊNDICE E – Modelo Matemático Completo

Este apêndice apresenta a formulação completa do modelo de Programação Linear Inteira utilizado na dissertação.

E.1 Variáveis de Decisão

$$x_{ij} = \begin{cases} 1 & \text{se o motorista } i \text{ realiza a tarefa } j, \\ 0 & \text{caso contrário.} \end{cases}$$

$h_i^{\text{dia}}, h_i^{\text{sem}}$: horas acumuladas diário e semanal.

E.2 Função Objetivo

$$\min Z = \sum_{i,j} c_{ij} x_{ij},$$

onde c_{ij} representa custos operacionais, distância, categoria ou penalidades.

E.3 Restrições

E.3.1 Atribuição Única

$$\sum_i x_{ij} = 1 \quad \forall j.$$

E.3.2 Limite de Condução Diária

$$h_i^{\text{dia}} = \sum_j d_j x_{ij} \leq 9 + e_i.$$

E.3.3 Janelas de Tempo

$$s_j x_{ij} \geq a_j, \quad s_j x_{ij} + d_j \leq b_j.$$

E.3.4 Compatibilidade Motorista–Tarefa

$$x_{ij} \leq \text{comp}(i, j).$$

E.3.5 Acúmulo Semanal

$$h_i^{\text{sem}} = \sum_j d_j x_{ij} \leq 56.$$

E.4 Modelo Final

Versão consolidada inclui todas as restrições regulatórias (CE 561/2006), limites de pausa, repouso e condução quinzenal, incorporadas por meio de estruturas lineares ou restrições complementares.

Anexos

ANEXO A – Resumo Estruturado do Regulamento (CE) n.º 561/2006

Este anexo apresenta um resumo das principais disposições do Regulamento (CE) n.º 561/2006, organizado em formato tabular para facilitar consulta durante a interpretação do modelo de escalonamento.

Limites de Condução

Tabela 5 – Limites de Condução

Parâmetro	Descrição
Condução diária	Máximo de 9 horas; permitido estender para 10 horas até 2 vezes por semana.
Condução semanal	Máximo de 56 horas.
Condução quinzenal (2 semanas)	Máximo de 90 horas acumuladas.

Pausas Obrigatórias

Tabela 6 – Regras de Pausa

Condição	Descrição
Pausa após condução contínua	45 minutos após no máximo 4h30 de condução.
Divisão da pausa	Permitido dividir em 15 min + 30 min.

Repousos Diários

Tabela 7 – Repousos Diários

Tipo	Descrição
Repouso regular diário	Mínimo 11 horas.
Repouso diário fracionado	3h + 9h (total mínimo 12h).
Repouso reduzido diário	Mínimo de 9 horas, até 3 vezes entre dois repousos semanais.

Repouso Semanal

Tabela 8 – Repousos Semanais

Tipo	Descrição
Repouso semanal regular	Mínimo de 45 horas.
Repouso semanal reduzido	Mínimo de 24 horas, compensação obrigatória em até 3 semanas.

ANEXO B – Glossário Técnico

PLI — Programação Linear Inteira: Modelo matemático onde todas as variáveis são inteiras, utilizado para problemas de decisão e alocação.

MIP — Mixed Integer Programming: Problemas que combinam variáveis inteiras e contínuas, resolvidos via métodos como Branch-and-Bound.

MIP Gap: Medida do quanto distante a solução corrente está da solução ótima, calculada como:

$$\text{Gap} = \frac{|Z^* - Z_{best}|}{Z^*}.$$

Presolve: Conjunto de transformações aplicadas pelo solver antes da otimização, reduzindo o tamanho do modelo.

Cutting Planes (Cuts): Restrições adicionais que eliminam regiões inviáveis da relaxação linear, acelerando convergência.

OR-Tools: Pacote de otimização do Google usado para resolver problemas de rota, scheduling e alocação.

Window Time (Janela de Tempo): Intervalo permitido para início ou término de uma tarefa.

Feasibility Pump: Heurística para encontrar soluções inteiras rapidamente em MIP.

ANEXO C – Fluxo Geral da Aplicação

Este anexo apresenta o fluxo geral da aplicação desenvolvida para implementação do modelo de Programação Linear Inteira aplicado ao escalonamento de motoristas sob o Regulamento (CE) n.º 561/2006.

O diagrama foi construído utilizando a linguagem PlantUML e ilustra, de forma abstrata e didática, todas as etapas principais da ferramenta: configuração dos parâmetros via interface, construção programática das variáveis e restrições, execução do solver CP-SAT e geração dos resultados analíticos.

Esse nível de detalhamento permite compreender a arquitetura e o funcionamento da solução sem expor o código-fonte propriamente dito, que é de natureza confidencial e constitui segredo comercial da organização.

A Figura 32 apresenta a visão geral do fluxo da aplicação.

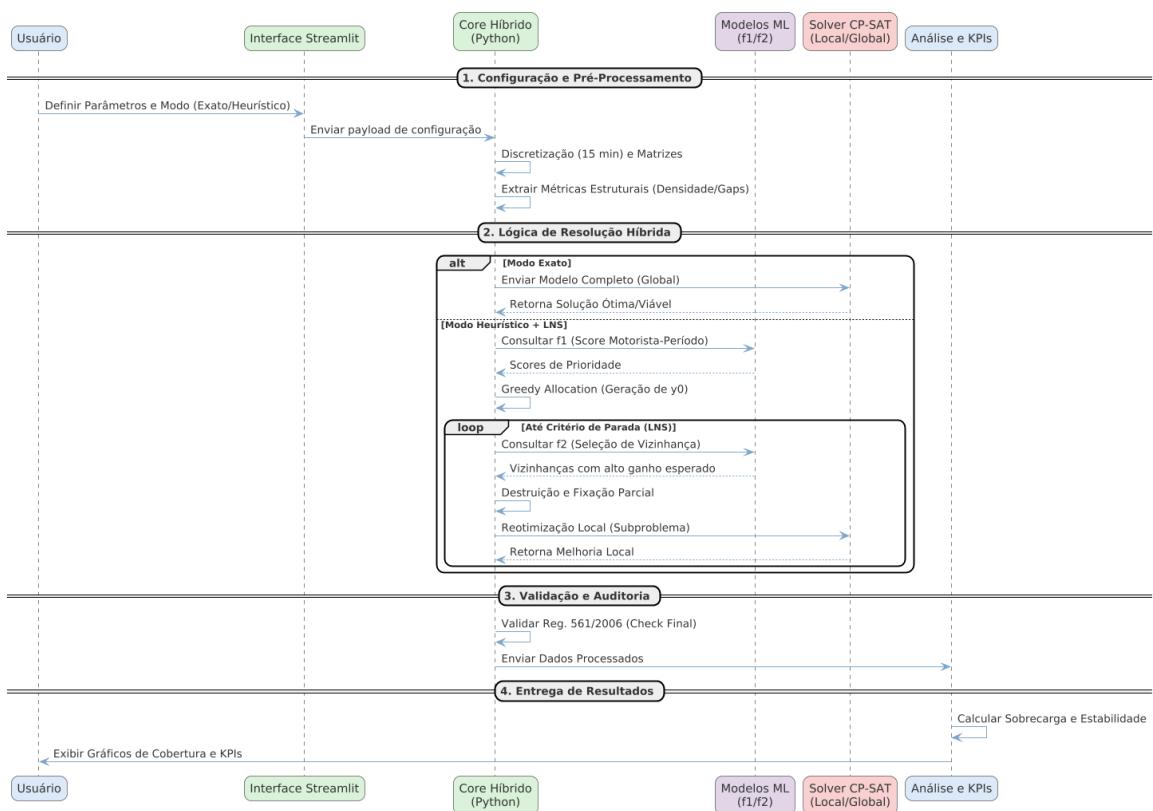


Figura 32 – Fluxo geral da aplicação desenvolvido em PlantUML.