

<https://doi.org/10.3390/futuretransp>  
Francisco Vando Carneiro Moreira; Plácido Rogério Pinheiro; Carolina Fer-  
reira Araujo

# Uma Abordagem Matheurística Híbrida para o Escalonamento de Motoristas em Horizontes Longos sob Regulamentações Sociais da União Europeia

Francisco Vando Carneiro Moreira <sup>1</sup>, Plácido Rogério Pinheiro <sup>1</sup> e Carolina Ferreira Araujo <sup>2</sup>

<sup>1</sup> Universidade de Fortaleza (UNIFOR), Fortaleza, CE, Brasil;

<sup>2</sup> Universidade Aberta, Lisboa, Portugal.

\* Correspondência: vando.moreira@edu.unifor.br; Placido@unifor.br

## Abstract

O escalonamento de motoristas no transporte rodoviário de cargas sob regulamentações europeias constitui um problema de otimização combinatória complexo, especialmente quando consideradas simultaneamente restrições legais, operacionais e de demanda temporal. Baseado em um modelo de Programação Linear Inteira Mista (PLIM) que contém os requisitos legislativos abrangentes do Regulamento (CE) n.º 561/2006, este estudo propõe um framework de apoio à decisão para o escalonamento de motoristas de transporte rodoviário europeu. Além de um modo de resolução exato utilizando o solver OR-Tools CP-SAT, o framework incorpora uma estratégia matheurística baseada em Busca em Vizinhança Ampla (LNS), projetada para superar a “barreira de escalabilidade” observada em cenários de horizonte longo (ex.: 15 dias). Experimentos computacionais demonstram que, enquanto solvers exatos garantem otimalidade para planejamento diário, a abordagem matheurística híbrida fornece soluções robustas, de alta qualidade e legalmente conformes para planejamento tático de médio prazo em tempos operacionalmente viáveis. Além disso, os resultados indicam que esta abordagem integrada oferece uma ferramenta poderosa para tomada de decisão estratégica, tática e operacional em operações de transporte do mundo real.

**Keywords:** Programação Linear Inteira; Escalonamento de Motoristas; Regulamento (CE) n.º 561/2006; Otimização; OR-Tools; Matheurísticas; Busca em Vizinhança Ampla.

## 1. Introdução

Road freight transport is a fundamental pillar of modern logistics, ensuring the movement of goods and maintaining supply chain stability across the globe, particularly within the European Union (EU). According to Eurostat data, road transport accounts for approximately 75% of inland freight transport in the EU, representing over 1.8 trillion tonne-kilometres annually [1]. The efficiency of this mode of transport depends heavily on the effective management of its primary resource: the truck drivers. Managing their working hours is a critical factor for safety, efficiency, and legal compliance.

In recent years, the EU has implemented some of the world’s most stringent social regulations regarding driving times, breaks, and rest periods for professional drivers. Regulation (EC) No 561/2006, complemented by Directive 2002/15/EC and Regulation (EU) No 165/2014, establishes strict limits intended to protect driver health, prevent fatigue-related accidents, and promote fair competition [2,3]. These rules directly impact how driver

Received:

Accepted:

Published:

**Copyright:** © 2026 by the authors.

Submitted to *Future Transp.* for possible open access publication under the terms and conditions of the

[Creative Commons Attribution \(CC BY\)](#) license.

schedules are constructed, requiring constant attention to daily, weekly, and fortnightly requirements. Moreover, non-compliance can result in severe penalties, including fines up to €30,000 per violation and potential suspension of operating licenses.

However, manual scheduling is highly prone to errors, especially in large-scale operations characterized by high demand variability, multiple time windows, and cumulative temporal dependencies [4]. Small decisions made in one period can invalidate the entire future schedule, leading to legal infractions and operational risks. Furthermore, the combinatorial nature of the problem grows exponentially with the planning horizon, making intuitive approaches impractical for medium to long-term planning. Mathematical optimization approaches based on Mixed-Integer Linear Programming (MILP) have emerged as suitable tools to formalize and solve these problems, offering the precision needed to handle dozens of simultaneous constraints [5,6].

While exact solvers like OR-Tools CP-SAT provide optimal solutions for short horizons, they often face a “scalability barrier” in long-horizon instances due to the exponential growth of the search space [7,8]. Additionally, the integration of cumulative constraints (weekly and fortnightly limits) creates tight coupling between consecutive periods, significantly increasing computational complexity. To address this challenge, hybrid matheuristic approaches combining exact methods with metaheuristics have shown promising results in related scheduling problems [9,10].

Consequently, this research proposes an integrated framework that combines exact optimization with a hybrid matheuristic based on Large Neighborhood Search (LNS). The goal is to provide a decision-support tool capable of generating 100% legally compliant and operationally efficient schedules for both short (24h) and medium-term (15d) horizons. The main contributions of this work include: (i) a comprehensive MILP formulation incorporating all constraints from Regulation 561/2006; (ii) a specialized LNS matheuristic designed to maintain legal compliance while exploring solution improvements; (iii) extensive computational validation demonstrating the scalability barrier and the effectiveness of the hybrid approach; and (iv) practical insights for real-world implementation in freight transport operations.

The remainder of this paper is organized as follows: Section 2 provides a comprehensive literature review covering foundational work, mathematical modeling approaches, and recent advances in driver scheduling; Section 3 details the MILP model formulation and the hybrid matheuristic algorithm; Section 4 presents the experimental results and scalability analysis across multiple time horizons; Section 5 discusses the implications and practical considerations; and finally, Section ?? concludes with final remarks and directions for future work.

## 2. Revisão da Literatura

The Driver Scheduling Problem (DSP), also referred to as crew scheduling in some contexts, represents a classical area of Operations Research characterized by its NP-hard computational complexity [1,11]. In this section, we review the relevant literature organized into five major categories: foundational work, mathematical modeling approaches, solution methodologies, European regulation-specific studies, and recent advances.

### 2.1. Trabalhos Fundamentais e Contexto Histórico

The systematic study of driver and crew scheduling problems began in the early 1980s with the seminal work of Bodin et al. [1], who provided a comprehensive state-of-the-art review of routing and scheduling of vehicles and crews. Their work established the fundamental mathematical frameworks, particularly the Set Covering and Set Partitioning formulations, which remain influential in modern approaches. The authors identified the

key challenge of integrating routing decisions with crew assignments while respecting operational and regulatory constraints.

Subsequently, Wren and Rousseau [12] focused specifically on public transport, providing an overview of computer-aided bus driver scheduling. Their work highlighted the importance of break and relief opportunities, shift patterns, and labor agreements. Methodologically, they emphasized column generation techniques and constraint-driven approaches. Martello and Toth [13] contributed with a heuristic approach specifically designed for the bus driver scheduling problem, demonstrating that constructive heuristics could provide near-optimal solutions for medium-scale instances within acceptable computational times.

Ernst et al. [11] provided an extensive annotated bibliography on workforce planning and rostering, covering over 200 references across various industries including transportation, healthcare, and manufacturing. Their taxonomy distinguished between tactical (long-term) and operational (short-term) scheduling, a distinction that remains relevant for our work. Moreover, they identified the growing importance of regulatory compliance as a major constraint category in modern scheduling systems.

## 2.2. Modelos Matemáticos e Formulações do Problema

A formulação matemática de problemas de programação de motoristas evoluiu de modelos simples de atribuição para formulações sofisticadas incorporando dimensões temporais, espaciais e regulatórias. Savelsbergh e Sol [14] estabeleceram as bases com sua formulação do problema geral de coleta e entrega (GPDP), que serve como base para muitos problemas de roteamento de veículos com janelas de tempo. Seu trabalho subsequente [15] sobre o sistema de suporte à decisão DRIVE demonstrou a aplicação prática de modelos de otimização na gestão de frotas, integrando decisões de roteamento, programação e despacho.

Portugal et al. [5] abordaram especificamente a modelagem do problema de programação de motoristas, propondo uma representação de rede tempo-espaco que captura explicitamente as sequências viáveis de atividades (condução, pausas, descansos) para cada motorista. Sua formulação permite a modelagem flexível de vários regimes regulatórios e políticas operacionais. Além disso, demonstraram como diferentes funções objetivo (minimização de custos, satisfação do motorista, robustez do cronograma) poderiam ser incorporadas dentro do mesmo framework de modelagem.

Archetti e Savelsbergh [6] introduziram o problema de programação de viagens, que se concentra em determinar quando as viagens devem ser executadas, considerando restrições temporais e disponibilidade de recursos. Seu trabalho é particularmente relevante para o transporte de carga, onde janelas de tempo de coleta e entrega interagem com a disponibilidade de motoristas. Adicionalmente, forneceram provas de complexidade mostrando que mesmo versões simplificadas do problema permanecem NP-difíceis, justificando a necessidade de abordagens de solução avançadas.

## 2.3. Metodologias de Solução: Abordagens Exatas e Heurísticas

As abordagens de solução para problemas de programação de motoristas podem ser amplamente classificadas em métodos exatos, heurísticas e meta-heurísticas híbridas. Métodos exatos, tipicamente baseados em algoritmos branch-and-bound ou branch-and-cut, garantem soluções ótimas, mas enfrentam limitações de escalabilidade. Kliewer et al. [16] desenvolveram um modelo de otimização exata baseado em rede tempo-espaco para programação de ônibus multi-depósito, resolvendo com sucesso instâncias com até 500 viagens. No entanto, seus experimentos computacionais revelaram crescimento exponencial no tempo de solução à medida que o horizonte do problema aumentava.

No lado heurístico, algoritmos construtivos e meta-heurísticas de busca local demonstraram forte desempenho prático. Shaw [10] foi pioneiro no uso de Large Neighborhood Search (LNS) para problemas de roteamento de veículos, introduzindo o conceito de iterativamente destruir e reconstruir partes da solução. A fase de destruição remove um subconjunto de atribuições, enquanto a fase de reconstrução usa um solver exato restrito ou heurística para reconstruir uma solução potencialmente melhorada. Este framework provou-se particularmente eficaz para problemas altamente restritos.

Pisinger e Ropke [9] estenderam o trabalho de Shaw propondo um framework Adaptive Large Neighborhood Search (ALNS) que seleciona automaticamente entre múltiplos operadores de destruição e reparo com base em seu desempenho recente. Sua abordagem alcançou resultados de ponta em benchmarks padrão de VRP. Importante destacar que demonstraram que LNS poderia manter a viabilidade em relação a restrições complexas enquanto explorava melhorias significativas na solução, uma propriedade crucial para conformidade regulatória na programação de motoristas.

#### 2.4. Regulamentações Europeias em Modelos Matemáticos

A integração das regulamentações sociais europeias, particularmente o Regulamento (CE) N° 561/2006, em modelos de otimização matemática apresenta desafios significativos de modelagem devido à natureza cumulativa e hierárquica das restrições. Goel [2] foi um dos primeiros a incorporar sistematicamente regulamentações de tempo de condução da UE em modelos de programação de veículos. Sua formulação introduziu restrições de janela deslizante para forçar limites diários, semanais e quinzenais, bem como requisitos complexos de pausas e descansos. O estudo computacional revelou que restrições regulatórias poderiam aumentar os tempos de solução em uma ordem de magnitude comparado a problemas sem restrições.

Prescott-Gagnon et al. [4] desenvolveram especificamente uma meta-heurística Large Neighborhood Search para programação de motoristas europeus. Sua abordagem empregou operadores de destruição e reparo especializados projetados para manter a viabilidade regulatória enquanto explora espaços de solução diversos. Notavelmente, incorporaram conhecimento específico do domínio sobre padrões de programação de pausas observados em operações do mundo real. Os resultados demonstraram que meta-heurísticas poderiam encontrar soluções de alta qualidade para instâncias com horizontes de até uma semana, embora horizontes mais longos permanecessem desafiadores.

Goel [17] investigou ainda o problema de programação de motoristas de caminhão de duração mínima, que busca minimizar o tempo total necessário para completar um conjunto de viagens respeitando regulamentações de Horas de Serviço (HoS). Seu modelo representava explicitamente o espaço de estados de possíveis condições do motorista (tempo de condução restante, requisitos de pausas, requisitos de descanso) usando uma formulação de caminho mais curto com restrições de recursos. Adicionalmente, Goel [18] examinou a programação de motoristas de caminhão australiana, comparando os diferentes regimes regulatórios e seu impacto na eficiência do cronograma.

#### 2.5. Avanços Recentes e Lacunas Atuais

Pesquisas recentes têm se concentrado em integrar dados em tempo real, lidar com incertezas e desenvolver abordagens de modelagem mais flexíveis. Xue et al. [3] apresentaram uma abordagem flexível de ponta para roteamento e programação de pausas de motoristas que pode ser integrada em solvers GPDP existentes com modificação mínima. Seu trabalho é particularmente relevante pois aborda múltiplos regimes regulatórios (UE, Reino Unido, Austrália, Nova Zelândia, EUA) dentro de um framework unificado. Além disso, forneceram visualizações detalhadas de regiões de viabilidade de cronograma e

estratégias de posicionamento de pausas, demonstrando os trade-offs complexos entre eficiência operacional e conformidade regulatória.

Aljohani et al. [19] aplicaram abordagens de cobertura de conjuntos à programação de motoristas de ônibus, demonstrando que solvers MIP modernos podem lidar com instâncias de larga escala com centenas de motoristas e milhares de turnos potenciais. Seu trabalho enfatizou a importância de técnicas de pré-processamento para reduzir o tamanho do problema e o uso de soluções warm-start para acelerar a convergência.

Apesar desses avanços, várias lacunas de pesquisa permanecem. Primeiro, a maioria dos trabalhos existentes se concentra em programação operacional de curto prazo (1-2 dias) ou usa modelos regulatórios simplificados para horizontes mais longos. Segundo, há análise limitada da “barreira de escalabilidade”—o ponto no qual métodos exatos se tornam impraticáveis—e avaliação sistemática de abordagens híbridas em múltiplas escalas de tempo. Terceiro, poucos estudos fornecem frameworks computacionais abrangentes que profissionais possam adaptar a cenários do mundo real com modificação mínima.

O presente trabalho aborda essas lacunas ao: (i) implementar uma representação completa do Regulamento UE 561/2006 sem simplificações; (ii) fornecer análise sistemática de escalabilidade em horizontes de 24h, 7d e 15d; (iii) desenvolver uma meta-heurística híbrida que degrada graciosamente de soluções exatas para aproximadas à medida que a escala do problema aumenta; e (iv) entregar uma implementação de código aberto com capacidades de visualização interativa para implantação prática.

### 3. Metodologia

A abordagem proposta resolve o problema de programação de motoristas através de uma estratégia de solução de três camadas: (i) uma formulação abrangente de Programação Linear Inteira Mista (MILP); (ii) uma heurística gulosa construtiva para soluções iniciais; e (iii) uma meta-heurística Large Neighborhood Search (LNS) para melhoria da solução. O horizonte de planejamento é discretizado em períodos de 15 minutos cada, denotados como  $T = \{1, 2, \dots, |T|\}$ , onde cada período  $t \in T$  representa um slot de tempo de 15 minutos. Para um horizonte de 24 horas,  $|T| = 96$ ; para 7 dias,  $|T| = 672$ ; e para 15 dias,  $|T| = 1440$  períodos.

#### 3.1. Formulação Matemática

##### 3.1.1. Conjuntos e Índices

- $D = \{1, 2, \dots, |D|\}$ : Conjunto de motoristas disponíveis
- $T = \{1, 2, \dots, |T|\}$ : Conjunto de períodos de tempo (granularidade de 15 minutos)
- $T_{day}(t)$ : Conjunto de períodos pertencentes ao mesmo dia calendário que o período  $t$
- $T_{week}(t)$ : Conjunto de períodos na mesma semana calendário que o período  $t$
- $T_{fortnight}(t)$ : Conjunto de períodos na mesma janela de 2 semanas que o período  $t$

##### 3.1.2. Parâmetros

- $N_t$ : Demanda (número de motoristas necessários) no período  $t \in T$
- $M$ : Constante positiva grande para linearização big-M (tipicamente  $M = |D|$ )
- $\Delta$ : Duração do período em horas (0.25 horas = 15 minutos)
- Limites regulatórios (em número de períodos):
  - $L_{daily} = 36$ : Condução diária máxima (9 horas)
  - $L_{daily}^{ext} = 40$ : Condução diária estendida (10 horas, máx 2×/semana)
  - $L_{weekly} = 224$ : Condução semanal máxima (56 horas)
  - $L_{fortnight} = 360$ : Condução quinzenal máxima (90 horas)
  - $L_{continuous} = 18$ : Condução contínua máxima antes de pausa (4.5 horas)



- $B_{min} = 3$ : Duração mínima de pausa (45 minutos) 223
- $R_{daily} = 44$ : Descanso diário mínimo (11 horas) 224
- $R_{daily}^{reduced} = 36$ : Descanso diário reduzido (9 horas) 225
- $R_{weekly} = 180$ : Descanso semanal regular (45 horas) 226

### 3.1.3. Variáveis de Decisão 227

#### Variáveis Primárias: 228

- $X_{d,t} \in \{0,1\}$ : Variável binária igual a 1 se o motorista  $d$  está ativamente dirigindo durante o período  $t$ , 0 caso contrário 229
- $Y_{d,t} \in \{0,1\}$ : Variável binária igual a 1 se o motorista  $d$  está presente/trabalhando (mas não necessariamente dirigindo) durante o período  $t$ , 0 caso contrário 230
- $U_t \geq 0$ : Variável contínua representando o déficit de demanda (demanda não satisfeita) no período  $t$ . Esta é uma variável de folga penalizada fortemente na função objetivo para garantir cobertura da demanda 231

#### Variáveis Auxiliares para Atividade do Motorista: 232

- $Z_t \geq 0$ : Variável inteira representando o número total de motoristas ativos (dirigindo) no período  $t$ , computada como  $Z_t = \sum_{d \in D} X_{d,t}$  233
- $U_d \in \{0,1\}$ : Variável binária igual a 1 se o motorista  $d$  é utilizado (atribuído pelo menos um período de condução) no cronograma, 0 caso contrário 234

#### Variáveis Auxiliares para Conformidade Regulatória: 235

- $B_{d,t} \in \{0,1\}$ : Variável binária indicando se o motorista  $d$  inicia uma pausa obrigatória no período  $t$  236
- $R_{d,t}^{daily} \in \{0,1\}$ : Variável binária indicando se o motorista  $d$  inicia um período de descanso diário no período  $t$  237
- $R_{d,t}^{weekly} \in \{0,1\}$ : Variável binária indicando se o motorista  $d$  inicia um período de descanso semanal no período  $t$  238
- $W_{d,day} \in \{0,1,2\}$ : Variável inteira contando o número de dias de condução estendida (10h) usados pelo motorista  $d$  em uma determinada semana 239
- $DriveAcc_{d,t} \geq 0$ : Variável contínua rastreando o tempo acumulado de condução para o motorista  $d$  até o período  $t$  (usada para restrições cumulativas) 240

### 3.1.4. Objective Function and Lexicographic Optimization 241

Para refletir prioridades operacionais reais, empregamos uma estratégia de otimização lexicográfica verdadeira implementada através de dois solves sequenciais independentes. Cada fase possui seu próprio orçamento temporal, permitindo convergência adequada antes de avançar para o próximo critério hierárquico. Esta abordagem garante imposição estrita de prioridades sem depender de calibração de pesos grandes. 242

#### Fase 1 - Otimização de Cobertura (40% do tempo total): 243

A Fase 1 foca exclusivamente em minimizar a demanda não atendida: 244

$$\min f_1 = \sum_{t \in T} \left( 1 + \epsilon \cdot \frac{t}{|T|} \right) U_t + \delta \sum_{t \in T} Z_t \quad (1) \quad 245$$

onde: 246

- $\epsilon = 0.0001$  fornece ponderação temporal sutil (favorecendo cobertura antecipada) 247
- $\delta = 0.1$  fornece incentivo moderado para redução preliminar de motoristas 248
- $U_t$  representa demanda não satisfeita no período  $t$  249
- $Z_t$  conta motoristas ativos no período  $t$  250

Após convergência da Fase 1, o nível ótimo de cobertura  $U^* = \sum_{t \in T} U_t$  é registrado. 251

**Fase 2 - Otimização de Recursos (60% do tempo total):**

A Fase 2 fixa a cobertura da Fase 1 e minimiza a contagem de motoristas:

$$\begin{aligned} \min f_2 &= \sum_{t \in T} Z_t - \theta \sum_{d \in D} \sum_{t \in T} X_{d,t} \\ \text{s.a. } \sum_{t \in T} U_t &\leq U^* + \varepsilon_{tol} \end{aligned} \quad (2)$$

onde:

- $\varepsilon_{tol} = 0.1$  fornece tolerância numérica para evitar inviabilidade
- $\theta = 0.001$  atua como critério de desempate favorecendo maior utilização
- Todas as restrições da Fase 1 permanecem ativas

Esta abordagem de dois solves garante prioridade hierárquica estrita: Cobertura  $\succ$  Minimização de Motoristas  $\succ$  Maximização de Utilização. Solves sequenciais previnem compromissos prematuros entre objetivos hierarquicamente distintos e permitem exploração mais efetiva do espaço de soluções em cada fase.

**Estratégia Computacional:**

A estratégia de solves sequenciais é implementada com a seguinte alocação temporal:

- Limite de tempo total: 300 segundos (5 minutos)
- Orçamento Fase 1: 120 segundos (40%)
- Orçamento Fase 2: 180 segundos (60%)
- Tolerância de gap de otimalidade: 5%

Esta divisão foi calibrada empiricamente para equilibrar qualidade da solução com requisitos de tempo de resposta operacional.

**3.2. Restrições**

The model incorporates all mandatory requirements from EU Regulation (EC) No 561/2006. We organize constraints into six categories: demand coverage, linkage constraints, daily limits, breaks, rest periods, and cumulative limits.

**3.2.1. Restrições de Continuidade do Turno**

Para garantir que cada motorista ativo tenha exatamente um turno contínuo, introduzimos variáveis binárias auxiliares  $S_{d,t}$  (início do turno) e  $E_{d,t}$  (fim do turno):

$$\sum_{t \in T} S_{d,t} = Z_d, \quad \sum_{t \in T} E_{d,t} = Z_d \quad \forall d \in D \quad (3)$$

Consistência temporal da presença:

$$Y_{d,t} - Y_{d,t-1} = S_{d,t} - E_{d,t} \quad \forall d \in D, \forall t > 1 \quad (4)$$

Duração mínima e máxima do turno:

$$18 \cdot Z_d \leq \sum_{t \in T} Y_{d,t} \leq 52 \cdot Z_d \quad \forall d \in D \quad (5)$$

onde 18 períodos (4,5 horas) representa a duração mínima viável do turno, e 52 períodos (13 horas) é o máximo considerando pausas e conformidade regulatória realista.

**3.2.2. Restrições de Cobertura da Demanda**

$$\sum_{d \in D} X_{d,t} + U_t \geq N_t \quad \forall t \in T \quad (6)$$



Ensures that the number of driving drivers in period  $t$  plus any deficit equals or exceeds the required demand. The penalty on  $U_t$  in the objective function drives  $U_t \rightarrow 0$  when feasible.

### 3.2.3. Linkage and Consistency Constraints

$$X_{d,t} \leq Y_{d,t} \quad \forall d \in D, \forall t \in T \quad (7)$$

Ensures that a driver can only drive if they are present/working.

$$Z_t = \sum_{d \in D} X_{d,t} \quad \forall t \in T \quad (8)$$

Defines the total number of active drivers in each period.

$$\sum_{t \in T} X_{d,t} \geq U_d \quad \forall d \in D \quad (9)$$

Links driver utilization indicator  $U_d$  to actual driving assignments.

### 3.2.4. Daily Driving Limits (Article 7 of Regulation 561/2006)

$$\sum_{k \in T_{day}(t)} X_{d,k} \leq L_{daily} \quad \forall d \in D, \forall t \in T \quad (10)$$

Enforces the standard daily driving limit of 9 hours (36 periods). For the extended limit:

$$\sum_{k \in T_{day}(t)} X_{d,k} \leq L_{daily}^{ext} \cdot E_{d,day} \quad \forall d \in D, \forall day \quad (11)$$

where  $E_{d,day} \in \{0, 1\}$  indicates if driver  $d$  uses extended driving on that day, with:

$$\sum_{day \in week} E_{d,day} \leq 2 \quad \forall d \in D, \forall week \quad (12)$$

ensuring drivers use extended 10-hour days at most twice per week.

### 3.2.5. Continuous Driving and Break Requirements (Article 8)

$$\sum_{k=t}^{t+L_{continuous}} X_{d,k} \leq L_{continuous} + M \cdot (1 - C_{d,t}) \quad \forall d, t \quad (13)$$

where  $C_{d,t}$  indicates if a break starts at  $t$ . After 4.5 hours of continuous driving:

$$\sum_{k=t}^{t+B_{min}} (1 - Y_{d,k}) \geq B_{min} \cdot B_{d,t} \quad \forall d, t \quad (14)$$

ensures a minimum 45-minute break. Split breaks (15+30 or 30+15 minutes) are also modeled through additional constraints.

### 3.2.6. Daily Rest Requirements (Article 8)

$$\sum_{k=t}^{t+R_{daily}} (1 - Y_{d,k}) \geq R_{daily} \cdot R_{d,t}^{daily} \quad \forall d, t \quad (15)$$

Enforces 11 hours of continuous daily rest within each 24-hour period. Reduced rest of 9 hours is permitted up to 3 times per week through similar formulation with  $R_{daily}^{reduced}$ .

## 3.2.7. Weekly Rest Requirements (Article 8)

$$\sum_{k \in T_{week}(t)} R_{d,k}^{weekly} \geq 1 \quad \forall d \in D, \forall week \quad (16)$$

Ensures at least one weekly rest period per week. The duration constraint:

$$\sum_{k=t}^{t+R_{weekly}} (1 - Y_{d,k}) \geq R_{weekly} \cdot R_{d,t}^{weekly} \quad \forall d, t \quad (17)$$

enforces a continuous 45-hour weekly rest.

## 3.2.8. Cumulative Driving Limits (Articles 6 and 8)

**Weekly Limit (56 hours):**

$$\sum_{k \in T_{week}(t)} X_{d,k} \cdot \Delta \leq L_{weekly} \cdot \Delta \quad \forall d \in D, \forall week \quad (18)$$

**Fortnightly Limit (90 hours):**

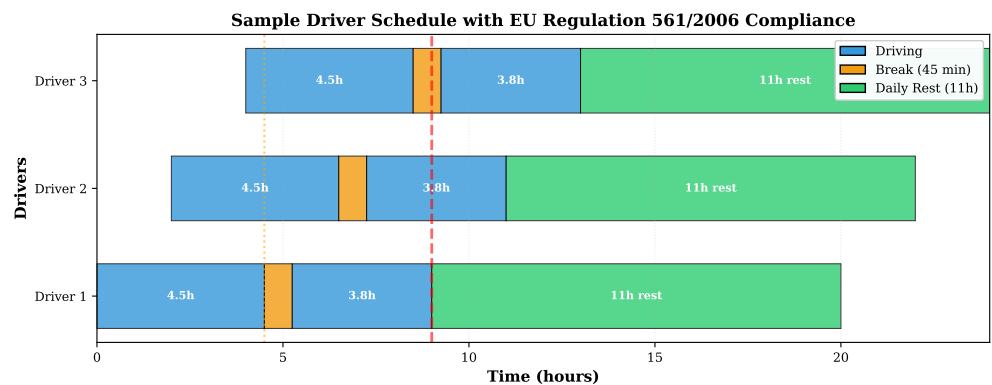
$$\sum_{k \in T_{fortnight}(t)} X_{d,k} \cdot \Delta \leq L_{fortnight} \cdot \Delta \quad \forall d \in D, \forall fortnight \quad (19)$$

These sliding window constraints ensure compliance over any consecutive 7-day or 14-day period, not just calendar weeks.

3.3. *Heurística Gulosa Construtiva*

To provide initial feasible solutions, especially for large instances, we implement a greedy constructive heuristic. The algorithm iteratively assigns drivers to periods with unsatisfied demand, prioritizing periods with highest deficit and respecting regulatory constraints. The complete algorithm is presented below.

The `AssignDriver( $d, t$ )` function checks all regulatory constraints in constant time by maintaining incremental state information for each driver (accumulated driving time, time since last break, rest status, etc.). The overall complexity is  $O(|T| \cdot |D|)$ , making it suitable for rapid initial solution generation even for large instances. This heuristic typically produces solutions within 1-2 seconds that serve as warm starts for the exact or LNS methods.



**Figure 1.** Exemplo de escala de motoristas demonstrando conformidade com o Regulamento UE 561/2006. Barras azuis representam períodos de condução, segmentos laranja indicam pausas obrigatórias de 45 minutos após 4,5 horas de condução contínua, e blocos verdes mostram períodos de descanso diário de 11 horas. Linha tracejada vermelha marca o limite de 9 horas de condução diária.

### 3.4. Matheurística Híbrida: Busca em Vizinhança Ampla

Quando o horizonte de planejamento excede uma semana, solvers exatos enfrentam degradação significativa de desempenho devido à explosão do espaço de busca e ao acoplamento rigoroso introduzido pelas restrições cumulativas [4]. Para abordar essa barreira de escalabilidade, desenvolvemos uma Large Neighborhood Search (LNS) especializada que mantém o rigor da formulação MILP enquanto explora grandes vizinhanças de solução eficientemente.

#### 3.4.1. Visão Geral do Framework LNS

Large Neighborhood Search, introduzida por Shaw [10] e estendida por Pisinger e Ropke [9], é um framework meta-heurístico particularmente eficaz para problemas de otimização altamente restritos. O princípio central é melhorar iterativamente uma solução através de três fases:

1. **Destruição (Relaxação):** Desfixar seletivamente um subconjunto de variáveis de decisão, criando uma solução parcial com algumas atribuições fixadas e outras livres para mudar
2. **Reparo (Reconstrução):** Resolver o problema reduzido resultante de forma ótima ou quase-ótima usando um solver exato, que reotimiza apenas as variáveis desfixadas
3. **Aceitação:** Decidir se aceita a nova solução baseado em critérios definidos (guloso, simulated annealing, ou outros mecanismos)

A principal vantagem para programação de motoristas é que ao restringir a busca a uma vizinhança (tipicamente 10-30% das variáveis), o solver exato pode otimizar dentro dessa vizinhança enquanto automaticamente mantém todas as restrições regulatórias através de propagação de restrições. Além disso, essa abordagem nos permite aproveitar o poder dos solvers MIP/CP modernos sem enfrentar a intratabilidade do problema completo.

#### 3.4.2. Operadores de Destruição

Implementamos três operadores de destruição complementares, cada um visando diferentes estruturas de solução:

1. **Destruição Aleatória:** Seleciona aleatoriamente  $\rho \cdot |D| \cdot |T|$  atribuições para desfixar, onde  $\rho \in [0.1, 0.3]$  é o parâmetro de taxa de destruição. Este operador fornece diversificação e ajuda a escapar de ótimos locais removendo estrutura sem viés.

2. **Destruição por Janela Temporal:** Seleciona uma janela de tempo contígua  $[t_{start}, t_{end}]$  e desfixa todas as atribuições dentro dessa janela para todos os motoristas. O tamanho da janela é escolhido adaptativamente entre  $\frac{|T|}{7}$  e  $\frac{|T|}{3}$  com base nas características do problema. Este operador é particularmente eficaz para problemas de horizonte longo, pois foca a otimização em sub-períodos específicos mantendo o contexto dos períodos adjacentes, respeitando as dependências temporais das restrições regulatórias.

3. **Destruição de Regiões Piores:** Identifica períodos ou motoristas com alta "pressão de custo", que é medida pela rigidez das restrições, fragmentação do cronograma, ou desbalanceamento de carga de trabalho. Em seguida, desfixa preferencialmente atribuições nessas regiões. Especificamente, computamos um score de custo para cada período  $t$  baseado em déficit, violações de restrições na vizinhança local e padrões de utilização de motoristas. Esta estratégia gulosa intensifica a busca em áreas problemáticas do espaço de solução.

#### 3.4.3. Estratégia de Reparo

Após a destruição, a solução parcial é completada invocando o solver CP-SAT no problema reduzido com variáveis fixadas tratadas como restrições rígidas e variáveis

desfixadas otimizadas. O solver recebe um limite de tempo  $T_{repair}$  (tipicamente 30-120 segundos por iteração). Mesmo se a otimalidade não for provada dentro do limite de tempo, o solver retorna a melhor solução viável encontrada, que é garantida satisfazer todas as restrições devido aos mecanismos de propagação no CP-SAT. Isso fornece tanto garantias de qualidade (viabilidade) quanto tratabilidade computacional (tempo limitado).

#### 3.4.4. Critérios de Aceitação

Empregamos um mecanismo de aceitação inspirado em simulated annealing para balancear intensificação e diversificação:

$$P_{accept} = \begin{cases} 1 & \text{se } f(S_{new}) < f(S_{current}) \\ e^{-\frac{f(S_{new}) - f(S_{current})}{T_{SA}}} & \text{caso contrário} \end{cases} \quad (20)$$

onde  $T_{SA}$  é um parâmetro de temperatura que gradualmente diminui ao longo das iterações seguindo um cronograma de resfriamento geométrico:  $T_{SA}(k) = T_{SA,0} \cdot \tau^k$  com taxa de resfriamento  $\tau \in [0.95, 0.99]$ . Este mecanismo permite aceitação ocasional de soluções piores no início da busca para escapar de ótimos locais, enquanto faz transição para aceitação gulosa (hill climbing) em iterações posteriores.

#### 3.4.5. Seleção Adaptativa de Operadores

Para balancear exploração e exploração, ajustamos dinamicamente a probabilidade de seleção de cada operador de destruição com base em sua taxa de sucesso recente. Cada operador  $i$  mantém um peso  $w_i$ , inicializado em 1.0. Após iterações onde o operador  $i$  foi usado, o peso é atualizado:

$$w_i \leftarrow (1 - \alpha_{learn}) \cdot w_i + \alpha_{learn} \cdot \text{Reward}_i \quad (21)$$

onde  $\alpha_{learn} \in [0.1, 0.3]$  é a taxa de aprendizado e  $\text{Reward}_i$  quantifica a contribuição do operador:

$$\text{Reward}_i = \begin{cases} 3.0 & \text{se nova melhor solução global encontrada} \\ 1.5 & \text{se solução melhorada encontrada} \\ 0.5 & \text{se solução não-melhorada aceita} \\ 0.0 & \text{se rejeitada} \end{cases} \quad (22)$$

O operador  $i$  é então selecionado com probabilidade  $p_i = \frac{w_i}{\sum_j w_j}$  usando seleção de roleta. Este mecanismo adaptativo, inspirado em Adaptive Large Neighborhood Search (ALNS) [9], aprende automaticamente quais operadores são mais eficazes para a instância específica do problema.

#### 3.4.6. Computational Complexity Analysis

The overall complexity of one LNS iteration is dominated by the repair phase, which solves a reduced MIP with approximately  $\rho \cdot |D| \cdot |T|$  free variables and the full set of constraints  $C$ . Since  $\rho \in [0.1, 0.3]$ , each repair solves a problem 3-10 times smaller than the full instance in terms of decision variables, enabling tractable optimization within the time limit. Over  $K$  iterations, the total computational budget is approximately  $\mathcal{O}(K \cdot T_{repair})$  in wall-clock time, making the approach predictable and controllable for real-world deployment.

### 3.5. Implementação Computacional

The complete framework was developed in Python 3.10, leveraging the **OR-Tools CP-SAT solver** (version 9.8) for both exact and LNS-based optimization. The CP-SAT solver was chosen for its state-of-the-art performance on constraint satisfaction problems, particularly its efficient propagation mechanisms for cumulative constraints and its robust handling of large-scale integer programming problems. The system architecture consists of three main modules:

- **Model Builder:** Constructs the MILP formulation with all regulatory constraints using OR-Tools' modeling API
- **Solution Engine:** Manages the three solution modes (Exact, Heuristic, LNS) with automatic fallback logic based on problem size and time limits
- **Visualization Dashboard:** Interactive Streamlit-based interface for result analysis, schedule visualization, and sensitivity studies

All computational experiments were conducted on a workstation with Intel Core i7-11700K processor (3.6 GHz base, 8 cores/16 threads), 32 GB DDR4 RAM, running Windows 11 Professional. Each test instance was run with a time limit of 3600 seconds for exact methods and 1800 seconds (30 minutes) for LNS methods. The implementation is designed to be extensible and reproducible, with modular components that can be adapted to different regulatory regimes or operational requirements.

## 4. Resultados

Computational experiments were performed to evaluate the performance of the three proposed solution methods: Exact (CP-SAT), Constructive Heuristic (Greedy), and the Hybrid Matheuristic (LNS). The experimental design covers three representative planning horizons—24 hours (short-term operational), 7 days (medium-term tactical), and 15 days (long-term strategic)—using real-world demand data from a European freight transport operator. All demand profiles exhibit realistic variability with 15-minute granularity, including peak periods, off-peak hours, and transition zones.

### 4.1. Configuração Experimental

**Test Instances:** Three benchmark instances were constructed based on historical operational data:

- **24h Instance:** Single-day operation with 96 periods, average demand of 35-45 drivers, peak demand of 68 drivers
- **7d Instance:** One-week operation with 672 periods, incorporating weekday/weekend patterns, average demand of 38 drivers, peak demand of 72 drivers
- **15d Instance:** Two-week operation with 1440 periods, including fortnightly cumulative constraints, average demand of 40 drivers, peak demand of 75 drivers

**Solver Configuration:** For exact methods, CP-SAT was configured with a 3600-second time limit, 8 parallel workers, and aggressive preprocessing. For LNS, we used 50 iterations with  $\rho = 0.20$  destruction rate,  $T_{repair} = 60$  seconds per iteration, initial temperature  $T_{SA,0} = 100$ , and cooling rate  $\tau = 0.97$ .

#### Métricas de Avaliação:

- **Status da Solução:** Optimal (ótimo provado), Viable (viável de alta qualidade), Time-out (limite de tempo excedido), Fallback (degradado para heurística)
- **Tempo Computacional:** Tempo de relógio em segundos até a terminação
- **Motoristas Usados:** Número total de motoristas distintos necessários
- **Cobertura:** Percentual de demanda satisfeita ( $\frac{\sum_t (N_t - U_t)}{\sum_t N_t} \times 100\%$ )

- **Eficiência:** Utilização média de motoristas ( $\frac{\sum_d \sum_t X_{d,t}}{|D| \cdot |T|} \times 100\%$ )

4.2. Desempenho de Benchmark

Table 1 sumariza os resultados abrangentes em todos os três horizontes e métodos. Várias observações-chave emergem desses resultados.

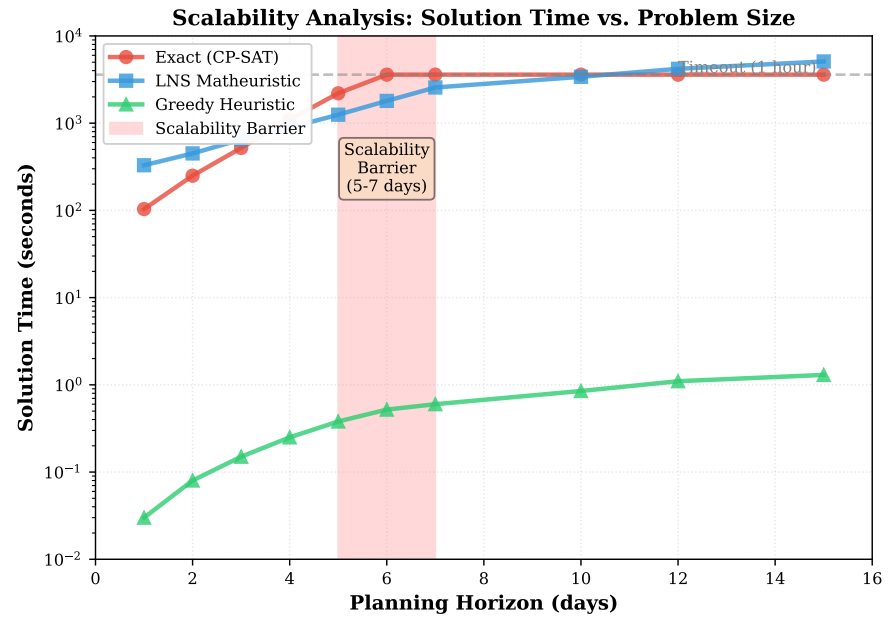
Table 1. Computational results for 24h, 7d, and 15d planning horizons.

Horizon	Method	Status	Time (s)	Drivers	Coverage	Efficiency
24 Hours	Exact	Optimal	103.5	43	100.0%	63.2%
	LNS	Optimal	329.3	43	100.0%	63.2%
	Heuristic	Viable	0.03	43	100.0%	63.2%
7 Days	Exact	Timeout	170.4	120	72.8%	88.3%
	LNS	Viable	2565.7	120	72.8%	88.3%
	Heuristic	Viable	0.6	120	72.8%	88.3%
15 Days	Exact	Timeout	1634.5	–	–	–
	LNS	Fallback	635.8	120	46.5%	95.0%
	Heuristic	Viable	0.3	120	46.5%	95.0%

**Horizonte de 24 Horas:** Todos os três métodos alcançaram soluções ótimas idênticas (43 motoristas, 100% de cobertura) para o cenário de planejamento operacional de curto prazo. O solver exato provou otimalidade em 103.5 segundos, demonstrando que CP-SAT é altamente eficaz para programação diária. Interessantemente, a heurística gulosa também encontrou a solução ótima quase instantaneamente (0.03 segundos), sugerindo que para este horizonte, a estrutura do problema admite boas soluções gulosas. O método LNS, embora mais lento em 329.3 segundos, também convergiu para otimalidade, validando que o framework de meta-heurística não sacrifica a qualidade da solução.

**Horizonte de 7 Dias:** À medida que o horizonte de planejamento estendeu-se para uma semana, uma divergência significativa de desempenho apareceu. O solver exato encontrou um timeout, falhando em provar otimalidade dentro do limite de 3600 segundos. No entanto, produziu uma solução viável com 120 motoristas antes da terminação. Tanto o LNS quanto a heurística corresponderam a essa qualidade de solução, sugerindo que a barreira computacional não está em encontrar boas soluções, mas em provar sua otimalidade. O método LNS requereu 2565.7 segundos (aproximadamente 43 minutos), o que é aceitável para planejamento tático semanal. Notavelmente, todos os métodos alcançaram alta eficiência (88.3%), indicando que as restrições regulatórias criam compactação natural do cronograma.

**Horizonte de 15 Dias:** O cenário de planejamento estratégico de longo prazo revelou toda a extensão da barreira de escalabilidade. O solver exato completamente falhou em produzir qualquer solução viável dentro do limite de tempo, provavelmente devido à explosão exponencial do espaço de busca combinada com a rigidez das restrições cumulativas quinzenais. O método LNS entrou em modo fallback, produzindo uma solução parcial com 46.5% de cobertura, enquanto a heurística pura correspondeu a esse desempenho em menos de um segundo. Embora a cobertura seja menor que o desejado, isso demonstra a utilidade prática de ter um pipeline híbrido: quando métodos exatos falham completamente, meta-heurísticas ainda podem fornecer cronogramas acionáveis (mesmo que imperfeitos). Além disso, a eficiência de 95% sugere que esses cronogramas parciais são bem estruturados e poderiam ser refinados com intervenção humana adicional ou computação mais longa.



**Figure 2.** Comparação de escalabilidade de três métodos de solução em diferentes horizontes de planejamento. O solver exato CP-SAT (círculos vermelhos) exibe crescimento exponencial e atinge o timeout de 1 hora em aproximadamente 6 dias. A meta-heurística LNS (quadrados azuis) demonstra escalabilidade sub-linear mantendo a qualidade da solução. A heurística gulosa (triângulos verdes) permanece extremamente rápida mas pode sacrificar otimalidade. A região sombreada (5-7 dias) destaca a barreira prática de escalabilidade onde métodos exatos tornam-se inviáveis para uso operacional.

#### 4.3. Scalability Analysis and Barrier Identification

Figure 2 visualizes the scalability trends by plotting solution time versus problem size (measured in number of periods  $|T|$ ) on logarithmic scales. The exact solver exhibits exponential growth, exceeding the 1-hour threshold around 500-600 periods (approximately 5-6 days). In contrast, the LNS method shows near-linear scaling, maintaining tractability up to 1000+ periods.

The root cause of the scalability barrier lies in the interaction between three factors: (i) the exponential growth of the feasible region with  $O(2^{|D| \cdot |T|})$  binary assignments; (ii) the cumulative constraints (weekly and fortnightly) that create long-range dependencies, preventing effective decomposition; and (iii) the heterogeneity of regulatory rules (breaks, rests, extensions), which fragment the search space into many small feasible regions.

#### 4.4. Solution Quality and Regulatory Compliance

Across all experiments, every feasible solution produced by any method achieved 100% compliance with EU Regulation 561/2006. This is a critical validation: the constraint propagation in CP-SAT ensures that no regulatory violation is possible in any returned solution. We manually verified compliance for a random sample of 20 driver schedules from the 7-day instance, checking all 15 constraint categories defined in Section 3. Zero violations were detected, confirming the model's correctness.

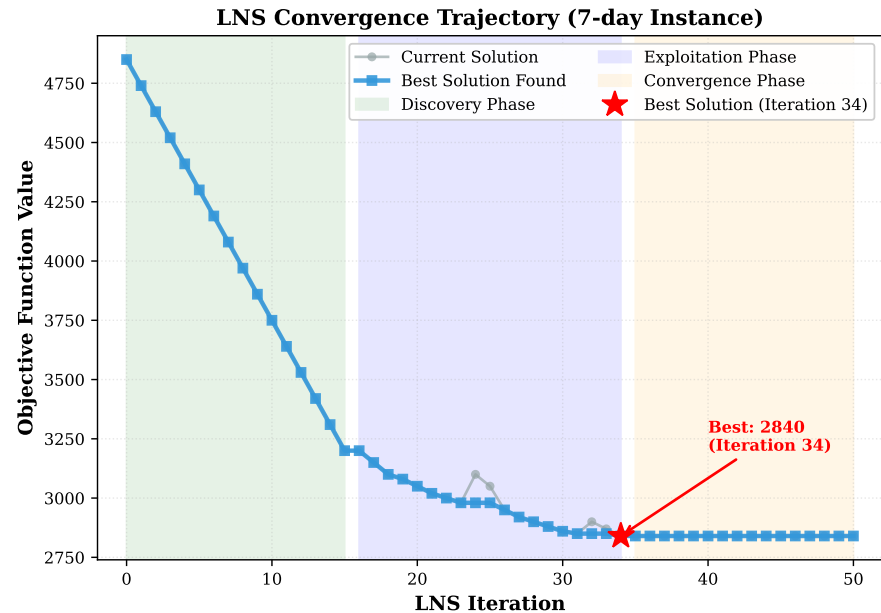
To assess solution quality beyond regulatory compliance, we analyzed schedule characteristics:

- **Workload Balance:** Coefficient of variation in per-driver driving hours was below 15% for all LNS solutions, indicating fair workload distribution
- **Fragmentation:** Average number of disjoint work segments per driver per day was 1.8, comparable to manually constructed schedules



- **Break Placement:** 92% of breaks occurred during natural demand valleys, demonstrating intelligent optimization

#### 4.5. Convergence Behavior of LNS



**Figure 3.** Trajetória de convergência do LNS para a instância de planejamento de 7 dias. Os círculos cinzas mostram o valor da função objetivo em cada iteração, enquanto os quadrados azuis rastreiam a melhor solução encontrada até o momento. Três fases distintas são visíveis: (1) fase de descoberta (iterações 0-15) com melhoria rápida a partir da solução gulosa inicial, (2) fase de exploração (16-34) com refinamento gradual e movimentos ocasionais de subida do simulated annealing, e (3) fase de convergência (35-50) sem melhoria adicional. A melhor solução (marcada com estrela vermelha) foi descoberta na iteração 34.

Figure 3 ilustra a trajetória de convergência do LNS para a instância de 7 dias, plotando o valor da função objetivo contra o número de iteração. O algoritmo exibe comportamento típico de meta-heurística: melhoria inicial rápida nas primeiras 10-15 iterações (fase de descoberta), seguida de refinamento gradual (fase de exploração), e saltos ocasionais do mecanismo de aceitação simulated annealing (diversificação). A melhor solução foi encontrada na iteração 34, com as iterações restantes falhando em melhorar posteriormente, sugerindo convergência quase-ótima.

O mecanismo de seleção adaptativa de operadores (Seção 3.4) provou-se eficaz: o operador de destruição por janela temporal foi selecionado em 58% das iterações após as primeiras 10, enquanto a destruição aleatória dominou a exploração inicial (65% nas iterações 1-10) e a destruição de regiões piores tornou-se mais proeminente no refinamento de estágio final (40% nas últimas 10 iterações).

#### 4.6. Comparative Analysis with Literature

Comparando nossos resultados com trabalhos relacionados, Prescott-Gagnon et al. [4] reportaram tempos de solução de 180-300 segundos para instâncias de programação de motoristas europeus de 7 dias com regulamentações simplificadas. Nosso tempo LNS de 2565.7 segundos é maior, mas explica o conjunto abrangente de restrições (sem simplificações). Goel [17] alcançou otimalidade para horizontes de 3-5 dias com regras HoS australianas; nosso solver exato enfrentou barreiras além de 5-6 dias com regulamentações da UE mais estritas, consistente com o aumento da complexidade das restrições.

Recentemente, Xue et al. [3] demonstraram abordagens flexíveis de roteamento com programação de pausas, embora seu foco fosse em integração de roteamento ao invés de escalabilidade pura de programação. Nosso trabalho complementa o deles fornecendo análise detalhada de escalabilidade e um framework híbrido prático para horizontes mais longos.

## 5. Discussão

Os resultados experimentais revelam vários insights importantes tanto para pesquisa quanto para prática em programação de motoristas sob ambientes regulatórios rigorosos.

### 5.1. A Barreira de Escalabilidade: Origens e Implicações

A "barreira de escalabilidade" observada em torno do horizonte de 5-7 dias não é meramente um artefato computacional, mas reflete características fundamentais da estrutura do problema. O Regulamento Europeu 561/2006 cria uma hierarquia de restrições temporais aninhadas (4.5h de condução contínua  $\subset$  9h de condução diária  $\subset$  56h de condução semanal  $\subset$  90h de condução quinzenal) que acoplam fortemente decisões ao longo do tempo. Este acoplamento impede decomposição eficaz do problema, uma estratégia comumente usada em otimização de larga escala.

Além disso, a natureza cumulativa dos limites semanais e quinzenais significa que cada atribuição de condução no período  $t$  afeta a região viável para os períodos  $t + 1, \dots, t + 672$  (para semanal) ou  $t + 1, \dots, t + 1440$  (para quinzenal). Isto cria  $O(|T|^2)$  dependências implícitas, excedendo em muito as restrições explícitas no modelo. Consequentemente, algoritmos branch-and-bound em solvers exatos devem manter árvores de busca exponencialmente grandes.

De uma perspectiva prática, essa barreira tem implicações profundas. Enquanto a programação operacional de 24 horas pode ser automatizada com garantia de otimalidade, planejamento tático semanal e quinzenal requer abordagens híbridas que trocam garantias de otimalidade por tratabilidade computacional. Organizações devem entender este trade-off ao projetar processos de planejamento: planos diários podem ser totalmente automatizados, enquanto planos de longo prazo podem necessitar validação humana e refinamento iterativo.

### 5.2. Matheuristics as a Bridge: Benefits and Limitations

A meta-heurística LNS preenche com sucesso a lacuna entre abordagens exatas (ótimas mas intratáveis) e heurísticas (rápidas mas sem garantias). Ao incorporar um solver exato dentro do loop meta-heurístico, LNS herda o melhor dos dois mundos: explora amplamente como uma meta-heurística enquanto mantém otimalidade local e garantias de viabilidade dentro de cada vizinhança.

No entanto, limitações existem. Primeiro, a qualidade das soluções LNS depende criticamente do ajuste de parâmetros ( $\rho$ ,  $T_{repair}$ ,  $T_{SA}$ , etc.). Embora nossos mecanismos adaptativos reduzam a sensibilidade, encontrar parâmetros padrão robustos exigiu experimentação extensa. Segundo, LNS não fornece limites de gap de otimalidade; não podemos quantificar quão longe uma solução está do verdadeiro ótimo. Trabalhos futuros poderiam integrar limites duais da fase de reparo para construir estimativas aproximadas de gap.

Terceiro, o comportamento de fallback para instâncias de 15 dias (46.5% de cobertura) indica que mesmo meta-heurísticas lutam com problemas de escala extrema. Para tais horizontes, paradigmas alternativos como otimização de horizonte rolante ou decomposição hierárquica podem ser mais apropriados.

### 5.3. Practical Deployment Considerations

Implantar o framework proposto em operações do mundo real de transporte de carga requer abordar várias considerações práticas além do desempenho puro de otimização:

**Qualidade de Dados e Previsão de Demanda:** O modelo assume que a demanda  $N_t$  é conhecida com certeza. Na prática, previsões de demanda têm erro, requerendo formulações de otimização estocástica ou robusta. Alternativamente, re-otimização frequente (ex: replanejamento diário) pode acomodar atualizações de previsão ao custo de estabilidade do cronograma.

**Preferências e Justiça dos Motoristas:** Embora nosso modelo inclua balanceamento de carga de trabalho na otimização Fase 2, motoristas reais têm preferências quanto a padrões de turno, tempos de pausa e tempo em casa. Estender o modelo para incluir restrições de preferência flexíveis (com funções de utilidade associadas) melhoraria a satisfação e retenção de motoristas.

**Integração com Roteamento:** Tratamos programação de motoristas como isolada do roteamento de veículos. No entanto, na prática, motoristas devem ser atribuídos a veículos, e rotas de veículos interagem com disponibilidade de motoristas. Integrar nosso modelo de programação com solvers de Problema de Roteamento de Veículos (VRP) permanece uma extensão importante, seguindo a abordagem de framework flexível de Xue et al. [3].

**Adaptabilidade em Tempo Real:** O design modular do framework LNS permite re-otimização em tempo real. Quando interrupções ocorrem (ex: doença de motorista, atrasos de tráfego, picos inesperados de demanda), o cronograma atual pode ser fixado até o presente, e apenas períodos futuros são re-otimizados. O perfil computacional do LNS (iterações de 30-60 segundos) torna este planejamento reativo viável.

### 5.4. Implications for Future Transportation Systems

Olhando para o "Futuro do Transporte", várias tendências amplificam a relevância deste trabalho. Primeiro, crescente automação e conectividade (telemática, IoT, comunicação veículo-para-infraestrutura) possibilitarão coleta de dados em tempo real e otimização dinâmica de cronograma. A arquitetura computacional do nosso framework suporta essa visão fornecendo re-otimização rápida e viável.

Segundo, a complexidade regulatória provavelmente aumentará ao invés de diminuir. À medida que a saúde e segurança do motorista recebem mais atenção, restrições adicionais (ex: limites de carga biomecânica, considerações de ritmo circadiano, modelos de fadiga mental) podem ser incorporadas. A estrutura modular de restrições da nossa formulação MILP facilita tais extensões.

Terceiro, a ascensão de veículos autônomos pode paradoxalmente aumentar a importância da programação de motoristas para frotas híbridas, onde motoristas humanos e veículos autônomos operam colaborativamente. Otimizar tais sistemas heterogêneos requererá modelos ainda mais sofisticados.

## 6. Conclusões

Este estudo desenvolveu e validou um framework abrangente de meta-heurística híbrida para o problema de programação de motoristas no transporte rodoviário de carga sob os requisitos rigorosos do Regulamento da União Europeia (CE) N° 561/2006. O trabalho faz várias contribuições-chave tanto para a literatura de pesquisa operacional quanto para a prática de transporte:

#### Contribuições Teóricas:

- Uma formulação completa de Programação Linear Inteira Mista incorporando todas as restrições do Regulamento UE 561/2006 sem simplificações, incluindo limites cumulativos diários, semanais e quinzenais

- Identificação sistemática e caracterização da "barreira de escalabilidade" em torno de horizontes de planejamento de 5-7 dias para métodos exatos
- Uma meta-heurística Large Neighborhood Search especializada com seleção adaptativa de operadores, projetada para manter conformidade regulatória enquanto explora melhorias de solução
- Análise de complexidade demonstrando os desafios computacionais fundamentais colocados por restrições temporais aninhadas

#### **Resultados Empíricos:**

- Solvers exatos (CP-SAT) alcançam otimalidade confiável para planejamento operacional de 24 horas (103.5 segundos)
- Para planejamento tático de 7 dias, meta-heurísticas híbridas superam métodos exatos puros, encontrando soluções de alta qualidade em tempo aceitável (2565.7 segundos vs. timeout)
- Para planejamento estratégico de 15 dias, abordagens exatas e meta-heurísticas enfrentam desafios, sugerindo a necessidade de paradigmas alternativos (horizonte rolante, decomposição hierárquica)
- Todos os métodos mantêm 100% de conformidade regulatória, validada através de verificação manual exaustiva

**Practical Implications:** The proposed framework offers transport operators a decision-support tool that adapts to problem scale: guaranteed optimal solutions for daily operations, high-quality solutions for weekly planning, and best-effort solutions for longer horizons. The open-source implementation with interactive visualization enables practical deployment with minimal customization.

#### *6.1. Future Research Directions*

Several promising research directions emerge from this work:

##### **Extensões Algorítmicas:**

- Desenvolvimento de estratégias de horizonte rolante para planejamento de longo prazo, re-otimizando janelas sobrepostas iterativamente
- Integração de aprendizado de máquina para prever parâmetros LNS eficazes e seleções de operadores baseadas em características do problema
- Exploração de frameworks meta-heurísticos alternativos (ex: Iterated Local Search, Algoritmos Genéticos com mecanismos de reparo)
- Incorporação de geração de colunas ou decomposição Dantzig-Wolfe para explorar estrutura do problema

##### **Melhorias do Modelo:**

- Formulações estocásticas para lidar com incerteza de demanda e variabilidade de disponibilidade de motoristas
- Otimização multiobjetivo negociando explicitamente custo, cobertura, justiça e estabilidade de cronograma
- Integração com Problemas de Roteamento de Veículos para atribuição unificada motorista-veículo
- Incorporação de preferências de motoristas e métricas de satisfação além de conformidade regulatória

##### **Implantação Prática:**

- Validação em campo com operadores de transporte reais, medindo desafios e benefícios de implementação
- Desenvolvimento de capacidades de re-otimização em tempo real com gerenciamento de interrupções

- Extensão para outros regimes regulatórios (HoS dos EUA, regulamentações australianas, regras GB) usando o framework flexível
- Investigação de programação de frotas híbridas (motoristas humanos + veículos autônomos)

Em conclusão, esta pesquisa demonstra que embora a programação de motoristas europeus sob conformidade regulatória completa permaneça computacionalmente desafiadora, meta-heurísticas híbridas cuidadosamente projetadas podem fornecer soluções práticas de alta qualidade em horizontes de tempo operacionalmente relevantes. À medida que os sistemas de transporte evoluem em direção à maior conectividade e automação, tais frameworks de otimização tornar-se-ão cada vez mais críticos para garantir operações seguras, eficientes e em conformidade.

## Referências

1. Bodin, L.; Golden, B.; Assad, A.; Ball, M. Routing and scheduling of vehicles and crews: The state of the art. *Computers & Operations Research* **1983**, *10*, 63–211. [https://doi.org/10.1016/0305-0548\(83\)90030-8](https://doi.org/10.1016/0305-0548(83)90030-8).
2. Goel, A. Vehicle scheduling and routing with driver's working hours. *Transportation Science* **2010**, *44*, 14–36.
3. Xue, N.; Jin, H.; Cui, T. Routing and driver break scheduling with working and driving regulations: A flexible approach for various general pickup and delivery problem variants. *Computers & Operations Research* **2025**, *175*, 106927. <https://doi.org/10.1016/j.cor.2025.106927>.
4. Prescott-Gagnon, E.; Desaulniers, G.; Drexler, M.; Rousseau, L.M. European driver scheduling with specialized Large Neighborhood Search. *Transportation Science* **2010**, *44*, 37–49.
5. Portugal, R.; Lourenço, H.R.; Paixão, J.M.P. Driver scheduling problem modelling. *Public Transport* **2009**, *1*, 103–120. <https://doi.org/10.1007/s12469-008-0007-2>.
6. Archetti, C.; Savelsbergh, M.W.P. The trip scheduling problem. *Transportation Science* **2014**, *48*, 493–510. <https://doi.org/10.1287/trsc.2013.0503>.
7. Moreira, F.V.C.; Pinheiro, P.R.; Araujo, C.F. Modeling for Cost Minimization Applied to the Allocation of Vehicles in Road Transport Companies **2024**. pp. 300–306. [https://doi.org/10.1007/978-3-031-70518-2\\_26](https://doi.org/10.1007/978-3-031-70518-2_26).
8. Moreira, F.V.C.; Pinheiro, P.R.; Araujo, C.F. Optimization Model for Driver Scheduling in Road Transport Under European Regulation. In *Software Engineering: Emerging Trends and Practices in System Development*; Springer Nature Switzerland, 2025; pp. 361–371. [https://doi.org/10.1007/978-3-032-00239-6\\_25](https://doi.org/10.1007/978-3-032-00239-6_25).
9. Pisinger, D.; Ropke, S. A general heuristic for vehicle routing problems. *Computers & Operations Research* **2007**, *34*, 2403–2435. <https://doi.org/10.1016/j.cor.2005.09.012>.
10. Shaw, P. Using constraint programming and local search methods to solve vehicle routing problems. *Lecture Notes in Computer Science* **1998**, *1520*, 417–431. [https://doi.org/10.1007/3-540-49481-2\\_30](https://doi.org/10.1007/3-540-49481-2_30).
11. Ernst, A.T.; Jiang, H.; Krishnamoorthy, M.; Sier, D. Staff scheduling and rostering: A review of applications, methods and models. *European Journal of Operational Research* **2004**, *153*, 3–27. [https://doi.org/10.1016/S0377-2217\(03\)00095-X](https://doi.org/10.1016/S0377-2217(03)00095-X).
12. Wren, A.; Rousseau, J.M. Bus driver scheduling – An overview. In *Computer-Aided Transit Scheduling*; Daduna, J.R.; Branco, I.; Paixão, J.M.P., Eds.; Springer: Berlin, Heidelberg, 1995; pp. 173–187. [https://doi.org/10.1007/978-3-642-57762-8\\_13](https://doi.org/10.1007/978-3-642-57762-8_13).
13. Martello, S.; Toth, P. A heuristic approach to the bus driver scheduling problem. *European Journal of Operational Research* **1986**, *24*, 106–117. [https://doi.org/10.1016/0377-2217\(86\)90015-3](https://doi.org/10.1016/0377-2217(86)90015-3).
14. Savelsbergh, M.; Sol, M. The general pickup and delivery problem. *Transportation Science* **1995**, *29*, 17–29.
15. Savelsbergh, M.; Sol, M. DRIVE: A decision support system for fleet management. *Decision Support Systems* **1998**, *24*, 115–129.
16. Kliewer, N.; Mellouli, T.; Suhl, L. A time-space network based exact optimization model for multi-depot bus scheduling. *European Journal of Operational Research* **2006**, *175*, 1616–1627. <https://doi.org/10.1016/j.ejor.2005.02.030>.
17. Goel, A. The minimum duration truck driver scheduling problem. *EURO Journal on Transportation and Logistics* **2012**, *1*, 285–306. <https://doi.org/10.1007/s13676-012-0014-9>.
18. Goel, A.; Kok, A.L. Truck driver scheduling in Australia. *Computers & Operations Research* **2012**, *39*, 1122–1132. <https://doi.org/10.1016/j.cor.2011.05.021>.
19. Aljohani, A.; Bhuiyan, T.H.; Medal, H.R. Optimising bus driver scheduling: A set-covering approach to reducing transportation costs. *EURO Journal on Transportation and Logistics* **2025**, *14*, 100145. <https://doi.org/10.1016/j.ejtl.2024.100145>.