UNIVERSIDADE FEDERAL DE JUIZ DE FORA

FACULDADE DE ENGENHARIA

CURSO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA

OTIMIZAÇÃO DA ALOCAÇÃO DE VAGÕES NO TRANSPORTE LOGÍSTICO FERROVIÁRIO INTEGRANDO PROGRAMAÇÃO INTEIRA MISTA E ALGORITMO GENÉTICO

MARIANA GUEDES MATTOS

JUIZ DE FORA

2025

UNIVERSIDADE FEDERAL DE JUIZ DE FORA

CURSO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA

OTIMIZAÇÃO DA ALOCAÇÃO DE VAGÕES NO TRANSPORTE LOGÍSTICO FERROVIÁRIO INTEGRANDO PROGRAMAÇÃO INTEIRA MISTA E ALGORITMO GENÉTICO

MARIANA GUEDES MATTOS

JUIZ DE FORA

FACULDADE DE ENGENHARIA

2025

OTIMIZAÇÃO DA ALOCAÇÃO DE VAGÕES NO TRANSPORTE LOGÍSTICO FERROVIÁRIO INTEGRANDO PROGRAMAÇÃO INTEIRA MISTA E ALGORITMO GENÉTICO

Trabalho Final de Curso apresentado ao Colegiado do Curso de Engenharia Elétrica da Universidade Federal de Juiz de Fora, como requisito parcial à obtenção do título de Engenheira Eletricista.

Área de conhecimento: Engenharia Elétrica

Orientador: Prof. Dr. Ivo Chaves da Silva Junior

Coorientador: Luiz Claúdio Antunes Costa

JUIZ DE FORA

FACULDADE DE ENGENHARIA

2025

OTIMIZAÇÃO DA ALOCAÇÃO DE VAGÕES NO TRANSPORTE LOGÍSTICO FERROVIÁRIO INTEGRANDO PROGRAMAÇÃO INTEIRA MISTA E ALGORITMO GENÉTICO

MARIANA GUEDES MATTOS

Trabalho Final de Curso submetido à banca examinadora constituída de acordo com o Artigo 9º do Capítulo IV das Normas de Trabalho Final de Curso estabelecidas pelo Colegiado do Curso de Graduação em Engenharia Elétrica, como parte dos requisitos necessários para a obtenção do grau de Engenheira Eletricista.

Aprovado em:

Por:

Orientador: PROF. DR. IVO CHAVES DA SILVA JUNIOR

Coorientador: LUIZ CLÁUDIO ANTUNES COSTA

PROF. DR. ANDRÉ LUIS MARQUES MARCATO

AGRADECIMENTOS

Agradeço, em primeiro lugar, à minha família, por estar ao meu lado em todos os momentos dessa caminhada.

À minha mãe, Regina Celi Guedes Mattos, por me ensinar a ter coragem, especialmente nos momentos em que as coisas não acontecem como esperamos. Ao meu pai, Luiz Carlos de Almeida Mattos, que, embora não esteja fisicamente presente para compartilhar esta conquista, permanece vivo em mim desde o meu nascimento. Aos meus irmãos, Ana Carolina Guedes Mattos e Matheus Guedes Mattos, por serem fontes constantes de inspiração, tanto pessoal quanto profissional.

Aos amigos que fiz ao longo da graduação e da vida, que celebraram comigo cada conquista e estiveram presentes nos desafios enfrentados nessa jornada.

Aos colegas de trabalho da MRS Logística S.A., pelo incentivo e apoio constantes, que foram fundamentais para que eu pudesse chegar até aqui.

Ao meu orientador, Prof. Ivo Chaves da Silva Junior, e ao meu coorientador, Luiz Cláudio Antunes Costa, pela orientação dedicada, paciência e valiosa contribuição para a realização deste trabalho. Agradeço ainda pela inspiração que representam como profissionais exemplares. Estendo também minha gratidão a todos os professores que contribuíram, de forma essencial, para minha formação acadêmica.

A todas essas pessoas, que foram parte fundamental e insubstituível nesta trajetória, deixo meu mais sincero agradecimento. Eu sou porque vocês são.

"Slow down you're doing fine. You can't be everything you wanna be before your time."

"Vá com calma, você está indo bem. Você não pode ser tudo o que deseja ser antes do seu tempo."

Billy Joel (tradução livre da autora)

RESUMO

Este trabalho apresenta um modelo híbrido de otimização para a alocação semanal de vagões plataforma na malha ferroviária da MRS Logística S.A. O objetivo principal é minimizar os cortes de volume (demanda não atendida) e, como objetivo secundário, reduzir o uso excessivo/ociosidade da frota por meio de uma penalidade α aplicada ao número de vagões. Inicialmente, empregou-se um modelo de Programação Inteira Mista (MIP), formulado em *Pyomo* e resolvido pelo solver open source CBC, garantindo soluções ótimas em instâncias de porte moderado. Para ampliar a escalabilidade e a flexibilidade de experimentação, desenvolveu-se um Algoritmo Genético (AG) com a biblioteca DEAP, calibrado via *grid search* e incluindo rotinas de reparo para assegurar a factibilidade das soluções.

Os dados de entrada (demanda semanal, disponibilidade por subtipo, capacidades úteis e tempos de ciclo) foram extraídos de planilhas operacionais e tratados em *Python*. No caso histórico de junho/2024 (cinco semanas), o MIP atendeu 465 383 t de um total de 541 410 t (85,96 %). Enquanto o melhor AG obteve 453 050 t (83,68 %) com 5 985 vagões. Os tempos de processamento permaneceram abaixo de alguns minutos em ambos os métodos. Esses resultados evidenciam que ferramentas de código aberto são capazes de resolver problemas logísticos complexos com qualidade próxima ao ótimo e custo computacional competitivo, fornecendo uma metodologia reprodutível e economicamente vantajosa para o planejamento operacional ferroviário.

Palavras-chave: logística ferroviária; alocação de vagões; programação inteira mista; algoritmo genético; otimização híbrida; *Pyomo*; DEAP.

LISTA DE ILUSTRAÇÕES

Figura 1 – Concessões ferroviárias no Brasil 12
Figura 2 – Fluxo de processamento (MIP à esquerda, AG à direita) 16
Figura 3 – Malha ferroviária sob concessão da MRS Logística S.A. (adaptado de (MRS Logística S.A., 2025)). 19
Figura 4 – Comparativo visual entre os sub-tipos de vagão plataforma Bobineiro e PET. 20
Figura 5 – Exemplos de Produtos Siderúrgicos. Fonte: elaboração própria. 21
Figura 6 – Demanda solicitada vs. atendida por semana. 35
Figura 7 – Taxa de atendimento (%) por semana. 36
Figura 8 – Taxa de atendimento por subtipo de vagão. 37

LISTA DE TABELAS

Tabela 1 – Comparativo conceitual entre MIP e AG 16
Tabela 2 – Indicadores logísticos e suas variáveis controláveis no presente estudo. . 17
Tabela 3 – Custo comparado de frete siderúrgico (São Paulo → Minas, 850km) . . 18
Tabela 4 – Principais sub-tipos de vagão plataforma na malha MRS 19
Tabela 5 – Classes de carga siderúrgica e sub-tipos de vagão mais adequados . . . 21
Tabela 6 – Comparação MIP versus AG em estudos selecionados 25
Tabela 7 – Exemplos da aba *Parâmetros Fluxos* 27
Tabela 8 – Disposição de vagões por categoria (extrato da aba *Disponibilidade*) . . 27
Tabela 9 – Disponibilidade semanal consolidada por tipo de vagão 27
Tabela 10 – Demanda semanal de três fluxos (aba *Volume*) 27
Tabela 11 – Conjuntos, parâmetros e variável de decisão do modelo 28
Tabela 12 – TU solicitada e atendida por semana 35
Tabela 13 – Subtipos mais demandados – TU acumulada nas cinco semanas 36
Tabela 14 – TU solicitada e atendida por semana – AG 41
Tabela 15 – Desempenho por subtipo de vagão – AG 41
Tabela 16 – Sensibilidade a α: efeito sobre cobertura e uso de vagões 42
Tabela 17 – Resumo global: comparação MIP *vs.* AG 42
Tabela 18 – Comparação por semana – TU solicitada *vs.* atendida 43
Tabela 19 – Comparação por subtipo de vagão – TU acumulada (5 semanas) 43
Tabela 20 – Resumo comparativo entre MIP e AG (mesma instância) 46
Tabela 21 – Sensibilidade a α: efeito sobre cobertura e uso de vagões (AG) 46

LISTA DE ABREVIATURAS E SIGLAS

AG	*Algoritmo Genético*
CO_2	Dióxido de Carbono
CBC	*Coin-OR Branch and Cut* (solver de PIM)
CPU	*Central Processing Unit*
DEAP	*Distributed Evolutionary Algorithms in Python*
DOI	*Digital Object Identifier*
ETL	*Extract, Transform, Load* (pipeline de dados)
GA	*Genetic Algorithm* (Algoritmo Genético)
GAP	Diferença relativa entre a solução obtida e o ótimo conhecido
GRNN	*Generalized Regression Neural Network*
MIP	*Mixed Integer Programming* (Programação Inteira Mista)
MRS	MRS Logística S.A.
OPEX	*Operational Expenditure* (custo operacional)
Pyomo	*Python Optimization Modeling Objects*
RAM	*Random Access Memory*
TU	*Toneladas Úteis* (capacidade de carga por vagão)
WSL2	*Windows Subsystem for Linux 2*
XML	*Extensible Markup Language*

LISTA DE SÍMBOLOS

F	Conjunto de fluxos de carga
S	Conjunto de semanas do horizonte de planejamento
T	Conjunto de sub-tipos de vagão
V	Conjunto de tipos de vagão ou disponibilidade (dependendo do contexto)
$x_{f,s}$	Número de vagões (qualquer tipo) alocados ao fluxo f na semana s
$x_{f,s,t}$	Número de vagões do subtipo t alocados ao fluxo f na semana s
$x^v_{f,t}$	Número de vagões do tipo v alocados ao fluxo f na semana t
$D_{f,s}$	Demanda de carga (t) do fluxo f na semana s
$\mathrm{TU}_{f,t}$	Capacidade útil de um vagão t no fluxo f (t/vagão)
$\mathrm{disp}_{v,t}$	Disponibilidade de vagões do tipo v na semana t
trips_f	Número máximo de viagens semanais permitido pelo ciclo do fluxo f
C_r	Tempo médio de ciclo (retorno) de cada vagão (dias)
$c_{f,t}$	Custo (ou penalidade) por tonelada não atendida no fluxo f, semana t
α	Coeficiente de penalização por demanda não atendida
γ	Coeficiente de penalização por transferência/excesso de alocação
$Z(x)$	Valor da função-objetivo MIP para a alocação x
$FIT(\mathbf{s})$	Função de aptidão do Algoritmo Genético para o indivíduo $\mathbf{s}$
$Z_{\geq 0}$	Conjunto dos números inteiros não negativos
$\max\{\cdot\}$	Operador máximo (retorna o maior valor do argumento)

SUMÁRIO

1	**INTRODUÇÃO**	**12**
1.1	Motivação	13
1.2	Objetivo	13
1.3	Metodologia	13
1.4	Eestrutura da Monografia	14
2	**REVISÃO DE LITERATURA**	**15**
2.1	Apresentação do Tema e Conceitos Básicos	15
2.2	Sazonalidade da Demanda no Transporte Ferroviário	17
2.3	Fundamentos da Logística Ferroviária	17
2.4	Contexto da Siderurgia no Brasil	17
2.5	Fluxos Siderúrgicos na MRS Logística S.A.	18
2.6	O Ciclo Operacional dos Vagões Plataforma	19
2.7	Tipos de Carga Siderúrgica e Relação com o Sub-tipo de Vagão	20
2.7.1	Influência da densidade e da geometria	21
2.7.2	Consequências para o modelo	22
2.8	Otimização da Alocação de Recursos em Ferrovias	22
2.8.1	Programação Inteira Mista na Otimização Logística	23
2.9	Meta-heurísticas e Algoritmos Genéticos na Logística	24
3	**METODOLOGIA**	**26**
3.1	Base de Dados e Parâmetros	26
3.2	Parâmetros Utilizados	28
3.3	Implementação Computacional e Construção do Modelo	28
3.4	Integração Pyomo/MIP em Ambiente Colaborativo	29
3.5	Ferramentas e Ambiente Computacional	29
3.5.1	Função-objetivo	29
3.5.2	Restrições	29
3.6	Métodos de Otimização	30
3.6.1	Implementação em `Pyomo`	30
3.6.2	Implementação do Algoritmo Genético	31
3.6.3	Representação do indivíduo	31
3.6.4	Função-objetivo (fitness)	31
3.6.5	Rotina de reparo (viabilidade aproximada)	32
3.6.6	Operadores e parâmetros evolutivos	32

3.6.7 Critério de parada e monitoramento . . . 33
3.6.8 Pós-processamento e exportação . . . 33
3.6.9 Sensibilidade ao parâmetro α . . . 33
3.6.10 Implementação em `DEAP` . . . 33
3.7 Paralelo com o Modelo MIP . . . 34
3.8 Ética e Reprodutibilidade . . . 34

4 RESULTADOS E DISCUSSÃO . . . 35
4.1 Atendimento de Demanda por Semana . . . 35
4.2 Discussão . . . 36
4.3 Desempenho por Subtipo de Vagão . . . 36
4.4 Fluxos com Maior Corte . . . 37
4.5 Causas Estruturais da Não Cobertura Total . . . 38
4.6 Indicadores de Utilização de Frota . . . 38
4.7 Sensibilidade ao Parâmetro α . . . 39
4.8 Paralelos com a Literatura . . . 40
4.9 Síntese Geral . . . 40
4.10 Resultados do Algoritmo Genético . . . 41
4.11 Sensibilidade ao parâmetro α . . . 41
4.12 Comparação entre o MIP e o Algoritmo Genético . . . 42
4.13 Discussão . . . 43

5 CONCLUSÃO E TRABALHOS FUTUROS . . . 45
5.1 Conclusões Principais . . . 45
5.2 Comparação MIP $\times$ AG . . . 46
5.3 Sensibilidade ao Parâmetro α . . . 46
5.4 Limitações do Estudo . . . 47
5.5 Recomendações Operacionais . . . 47
5.6 Trabalhos Futuros . . . 47
5.7 Considerações Finais . . . 48

REFERÊNCIAS . . . 60

1 INTRODUÇÃO

Segundo Agência de Notícias da Indústria (2024), a distribuição modal de transporte de cargas no Brasil atualmente se divide entre rodovias (62,0 %), ferrovias (19,1 %), modal aquaviário (14,6 %), dutoviário (3,6 %) e aéreo (0,1 %). Embora o modal rodoviário domine a matriz, o setor ferroviário, com uma malha de 7 856,8 km, apresenta vantagens em eficiência logística e sustentabilidade (Agência Nacional de Transportes Terrestres (ANTT), 2024).

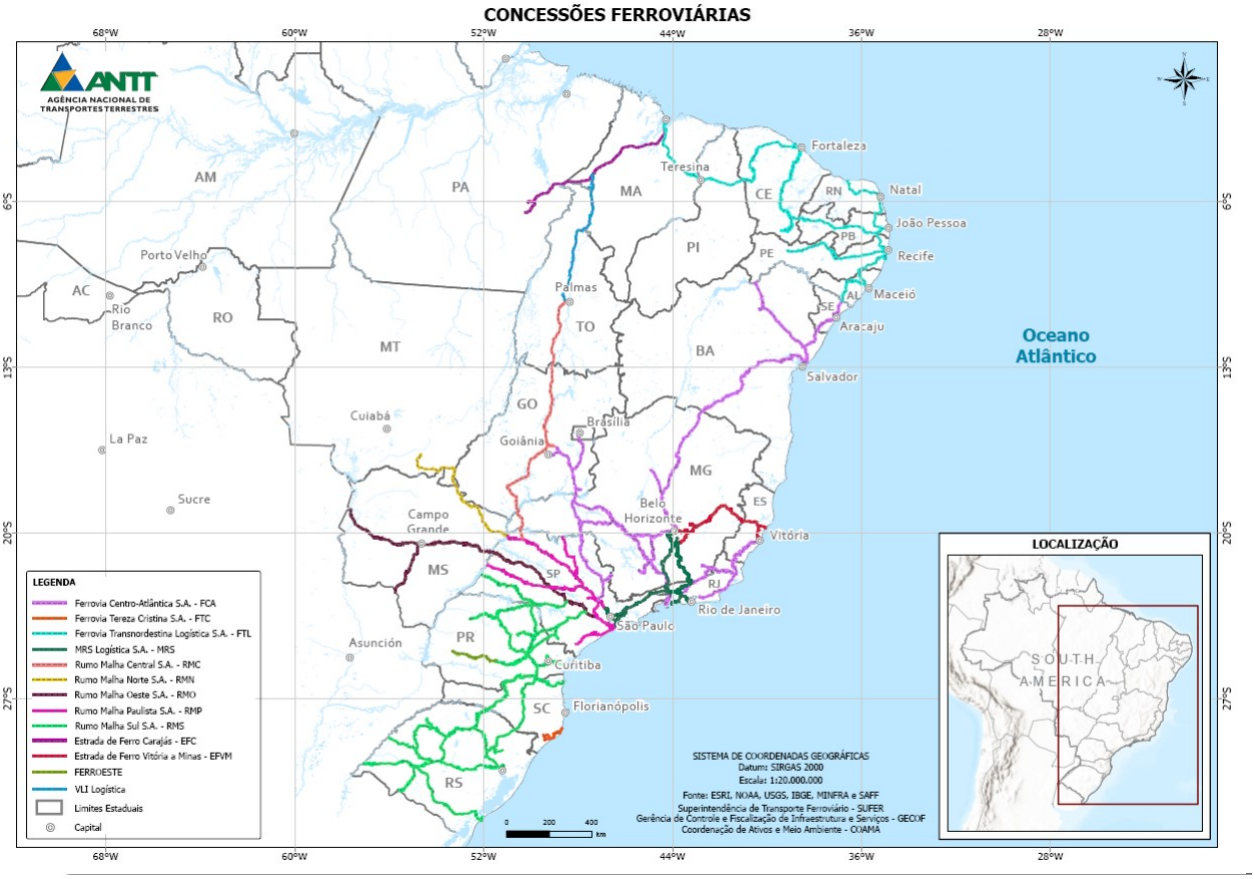


Figura 1 – Concessões ferroviárias no Brasil

Fonte: (Agência Nacional de Transportes Terrestres (ANTT), 2024)

O transporte ferroviário de cargas enfrenta variabilidade sazonal intensa, o que torna o planejamento da alocação de vagões especialmente desafiador. Variações mensais na demanda, somadas a ciclos de retorno e limites de disponibilidade de vagões, podem provocar tanto subutilização de ativos quanto cortes no volume transportado. Nesse contexto, métodos de otimização baseados em meta-heurísticas, como Algoritmos Genéticos (AG), surgem como ferramenta promissora para encontrar boas soluções em espaços de busca de alta complexidade, sem a necessidade de linearizar todas as restrições.

1.1 Motivação

A sazonalidade da demanda e a diversidade de tipos de carga exigem modelos flexíveis capazes de explorar múltiplas combinações de alocação de vagões por fluxo e período. A aplicação de um AG permite:

- Lidar com restrições não lineares (ciclos de retorno, penalidades de atraso, preferências de tipo de vagão);
- Obter soluções de alta qualidade em tempo computacional controlado;
- Ajustar facilmente operadores e parâmetros para cenários futuros.

Assim, aprimorar o planejamento sazonal de alocação de vagões é fundamental para maximizar o atendimento da demanda e reduzir custos logísticos no transporte ferroviário.

1.2 Objetivo

Este trabalho tem como objetivos principais:

- Desenvolver um modelo de otimização por Algoritmo Genético para alocação sazonal de cargas em malha ferroviária;
- Implementar em Python, com suporte à leitura de parâmetros de demanda, disponibilidade e tipologia de vagões;
- Validar a metodologia em estudo de caso real, comparando o desempenho do AG com solução exata de referência.

1.3 Metodologia

No Capítulo 2 realiza-se revisão de literatura sobre o uso de métodos de otimização na logística, especialmente em ferrovias, e fundamentos de Algoritmos Genéticos. O Capítulo 3 descreve:

1. A base de dados e parâmetros do problema (demanda sazonal, capacidades, tempos de ciclo);
2. A formulação da função fitness e penalizações associadas;
3. Detalhes de implementação com a biblioteca DEAP em Python e configuração dos operadores genéticos.

Em seguida, no Capítulo 4 apresentam-se resultados computacionais, análise de convergência do AG e comparação com modelo de Programação Inteira Mista (Pyomo/CBC). Finalmente, o Capítulo 5 discute conclusões, limitações e propostas de trabalhos futuros.

1.4 Eestrutura da Monografia

- **Capítulo 2** – Revisão de Literatura: conceitos de sazonalidade e meta-heurísticas aplicadas à logística.
- **Capítulo 3** – Metodologia: modelagem do problema e implementação do MIP e Algoritmo Genético.
- **Capítulo 4** – Resultados: experimentos, métricas de desempenho e comparação com abordagem exata.
- **Capítulo 5** – Conclusões e Perspectivas: síntese dos achados e direções para trabalhos futuros.

2 REVISÃO DE LITERATURA

2.1 Apresentação do Tema e Conceitos Básicos

A dinâmica do transporte logístico, especialmente o ferroviário, enfrenta desafios que vão além da simples movimentação de cargas. Um dos principais é a *sazonalidade* da demanda: flutuações periódicas ao longo das semanas do mês, impulsionadas por ciclos industriais, condições climáticas e datas sazonais, que geram picos e vales de necessidade de transporte. É necessário usar métodos que garantam a melhor *eficiência operacional* com menor uso dos ativos possível.

Meta-heurísticas são ferramentas poderosas neste contexto. Ao contrário de abordagens exatas, que buscam o ótimo global por meios determinísticos, as meta-heurísticas exploram o *espaço de soluções* de forma inteligente, aceitando soluções quase ótimas em troca de redução no tempo de processamento. Entre elas, os *Algoritmos Genéticos* (AG) destacam-se pela analogia ao processo evolutivo: populações de soluções sofrem ciclos de seleção, cruzamento e mutação, incentivando a sobrevivência das configurações mais aptas.

Para configurar um AG, definem-se:

- **Representação de Solução:** vetor de inteiros, em que cada elemento indica quantos vagões são alocados a um fluxo f na semana s;
- **Função de Aptidão (Fitness):** avalia a qualidade da alocação, combinando o aproveitamento da capacidade (toneladas úteis por vagão) e penalidades por demanda não atendida ou excesso de alocação;
- **Operadores Genéticos:** regras de *crossover*, *mutação* e *seleção*, como crossover de dois pontos, mutação uniforme e seleção por torneio.

Enquanto os AG buscam boas soluções por mecanismos evolutivos estocásticos, a MIP segue um caminho determinístico: o problema é formulado como um sistema de equações e inequações lineares com variáveis inteiras, resolvido por *branch–and–bound* e cortes válidos. A cada nó da árvore de busca o solver resolve um relaxamento linear; quando o *gap* entre o limite inferior e o melhor valor inteiro cai abaixo do tolerado, obtém-se o *ótimo global* para o modelo. Para configurar uma MIP definem-se:

- **Representação de Solução:** matriz de inteiros $x_{f,s,v}$ na qual cada célula indica quantos vagões do tipo v são alocados ao fluxo f na semana s;
- **Função-Objetivo:** combinação linear que minimiza o corte de volume e penalidades por excesso ou desbalanceamento de frota, ponderadas por coeficientes α e γ;

- **Conjunto de Restrições:** *i)* demanda não ultrapassada, *ii)* disponibilidade semanal de cada subtipo, *iii)* limite de viagens por fluxo em função do tempo de ciclo, todas expressas como inequações lineares;
- **Mecanismo de Solução:** algoritmo branch–and–bound com geração automática de cortes (Gomory e flux balance), opção de execução paralela (`-threads 4`) e critério de parada `mipgap` $= 10^{-4}$.

Dessa forma, podemos descrever a Programação Inteira Mista como uma abordagem exata que modela a alocação por meio de variáveis inteiras e restrições lineares, garantindo a solução ótima global. Contudo, seu tempo de CPU cresce exponencialmente em conjunto com a complexidade do problema. Enquanto que para os Algoritmos Genéticos possuímos a meta-heurística populacional que explora o espaço de busca de forma estocástica, trocando a garantia de otimalidade por tempos de CPU previsíveis e facilidade de incorporar restrições não lineares.

A Tabela 1 resume as principais particularidades.

Tabela 1 – Comparativo conceitual entre MIP e AG

Aspecto	**MIP (modelo exato)**	**AG (meta-heurística)**
Representação	Variáveis inteiras $x^v_{f,t}$ em um sistema de equações lineares	Cromossomo: vetor de inteiros $(x^v_{f,t})$
Exploração	Branch–and–Bound, cortes válidos; garante ótimo global	Seleção, crossover, mutação; converge a quase-ótimos
Tempo de CPU	Cresce exponencialmente com $\|F\| \times \|T\|$; sensível a big M	Aproximadamente linear; escalável
Manejo de restrições não lineares	Necessita linearização ou relaxamentos	Pode ser tratado via penalidades/reparo na fitness

A Figura 2 ilustra o fluxo lógico de cada abordagem.

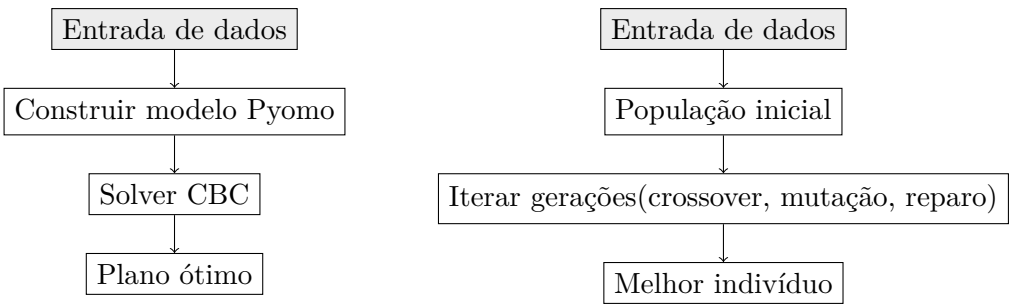


Figura 2 – Fluxo de processamento (MIP à esquerda, AG à direita)

Apesar de compartilharem o mesmo pré-processamento de dados os modelos divergem no mecanismo de busca por soluções: o MIP percorre uma árvore de Branch–and–Bound

até fechar o *gap*, enquanto o AG evolui uma população de alocações, aceitando soluções sub-ótimas em troca de menor tempo computacional.

Nos próximos capítulos serão explorados ambos os métodos, explicitando suas particularidades.

2.2 Sazonalidade da Demanda no Transporte Ferroviário

A sazonalidade no transporte ferroviário reflete padrões complexos de produção e logística. Zhang et al. (2016) integraram Redes Neurais Generalizadas (GRNN) a AG para prever volumes mensais com maior precisão, reduzindo erros em até 15% em comparação a métodos estatísticos convencionais (Zhang; Li; Zhou, 2016). Esses resultados mostram a importância de combinar previsão de séries temporais com otimização para capturar picos de safra agrícola ou demanda industrial.

Estudos em transporte rodoviário e marítimo ajustaram rotas e frequências conforme sazonalidade, obtendo redução de custos operacionais e tempo de trânsito.

2.3 Fundamentos da Logística Ferroviária

Tabela 2 – Indicadores logísticos e suas variáveis controláveis no presente estudo.

Indicador	**Equação**	**Fonte de Dados**
Ciclo (CT)	$T_{viagem} + T_{carga} + T_{desc} + T_{pátio}$	Histórico MRS 2024
Demanda Atendida (DA)	$\frac{tonpendentes}{demandatotal}$	Planilha de demanda semanal
Utilização média (U)	$\frac{\sum TU}{\sum capacidadekm}$	Planilha de Utilização

A sazonalidade pode gerar variações semanais de até 40 %, exigindo políticas de alocação dinâmicas (Cordeau; Laporte; Mercier, 2000). Nos dados estudados fornecido pela MRS Logística S.A., janelas portuárias e necessidade de atendimento dos alto fornos concentram demanda em períodos específicos.

2.4 Contexto da Siderurgia no Brasil

O setor siderúrgico brasileiro é um pilar da economia. Em 2023, o país produziu 35,1 milhões de toneladas de aço bruto, ocupando o 10º lugar global (Anuário Estatístico do Aço; 2025).

A cadeia siderúrgica também apresenta sazonalidade:

- *Obras de infraestrutura* elevam demanda por bobinas no segundo semestre;

- *Mercado da construção* ajusta pedidos de vergalhões conforme ritmo econômico;
- *Exportações* dependem de janelas portuárias e cotações internacionais.

Embora o frete ferroviário seja mais atrativo, levando em conta sua emissão de CO_2 78% menor que o modal rodoviário, nem todo aço embarca nos trilhos. A principal barreira é a restrição de frota. Segundo a ANTF, o número de de vagões plataforma dedicados a siderurgia caiu no ano de 2024, enquanto a produção de bobinas cresceu. Essa mostra a importância da construção de trabalho que possuem como objetivo central a alocação eficiente de vagões para evitar gargalos em aciarias, reduzir custos de espera em pátios e alinhar embarques. A Tabela 3 resume indicadores médios de 2023 que comparam o modal ferroviário e rodoviário.

Tabela 3 – Custo comparado de frete siderúrgico (São Paulo → Minas, 850km)

Modal	R$ /t·km	Emissão CO_2 (kg/t·km)
Rodoviário	0,305	0,123
Ferroviário	0,196	0,027

Fonte: CNT (2024), ANTT (2024).

2.5 Fluxos Siderúrgicos na MRS Logística S.A.

A ferrovia administrada pela MRS Logística S.A. cobre 1 643km, equivalentes a aproximadamente 6% da extensão ferroviária nacional, mas responde por cerca de 33% da produção de carga do País. Atualmente, a MRS detém cerca de 20% da frota ferroviária nacional, com 20 mil vagões e 650 locomotivas.— proporção que evidencia o nível de produtividade da malha e sua importância sistêmica (MRS Logística S.A., 2025). Graças à sua posição geográfica, o traçado conecta os principais polos produtores (Quadrilátero Ferrífero e Médio Paraíba), mercados consumidores do Sudeste e cinco grandes portos (Rio de Janeiro, Itaguaí, Sepetiba, Santos e Vitória), funcionando como corredor logístico de minerais, aço e contêineres.

Em 2024, o Brasil registrou **34,1Mt de aço bruto** e **2,1Mt de ferro-liga**, dos quais *29%* saíram diretamente de Minas Gerais (Instituto Aço Brasil, 2025). Os principais corredores de transporte:

- **Fluxo MG ES** — abastecimento de pelotas e sínter para alto-fornos do Complexo Tubarão (25 Mt/ano);
- **Fluxo MG/RJ SP** — envio de bobinas a quente e vergalhões para exportação via Santos (14 Mt/ano);



Figura 3 – Malha ferroviária sob concessão da MRS Logística S.A. (adaptado de (MRS Logística S.A., 2025)).

- **Fluxo RJ MG** — retorno de sucata ferrosa e insumos auxiliares para aciarias mineiras (3 Mt/ano).

2.6 O Ciclo Operacional dos Vagões Plataforma

Os vagões plataforma são utilizados no transporte de carga geral principalmente de fluxos siderúrgicos e de container. Essa frota possui algumas particularidades quanto a cargas e descargas dos produtos por possuir processos diferentes para cada tipo de produto siderúrgico. A frota analisada no presente trabalho é composta, majoritariamente, por vagões do tipo plataforma. Quatro sub-tipos concentram mais de 90 % do estoque disponível:

Tabela 4 – Principais sub-tipos de vagão plataforma na malha MRS

Sub-tipo	Finalidade típica	Capacidade (t)	Compr. (m)
LISO	Chapas grossas, perfis, cargas gerais	65–75	18.4
BOBINEIRO	Bobinas a quente/frio (berços em "V")	60–68	18.4
MULTIUSO	Siderúrgicos leves, sucata compactada	68–72	18.4

Fonte: especificações técnicas MRS (2024).

A Plataforma Lisa é um Vagão com piso metálico reforçado e laterais rebatíveis; aceita sobrecarga pontual de até 80t, sendo o mais versátil para produtos laminados planos.

A Plataforma Bobineiro é equipada com berços em "V" e correntes de ancoragem, atende à NBR 16201 para estabilizar bobinas de até 32t cada.

A Plataforma Multiuso possui estrutura aberta, sem berços fixos, indicada para sucata prensada ou embarques ocasionais.

Cada subtipo é utilizado em determinado fluxo que possui particularidades ou especificações necessárias para o transporte de forma segura. Essa é uma das limitações contempladas no presente trabalho.



(a) Vagão plataforma **bobineiro**



(b) Vagão plataforma **PET**

Figura 4 – Comparativo visual entre os sub-tipos de vagão plataforma Bobineiro e PET.

De forma geral o vagão plataforma de acordo com o *Relatório de Desempenho do Setor Ferroviário 2024* publicado pela Agência Nacional de Transportes Terrestres (ANTT), percorre em média, 620 km em cada ciclo completo de operação na malha da MRS Logística, englobando o trajeto de ida e volta entre origem e destino, além das manobras de pátio (Agência Nacional de Transportes Terrestres (ANTT), 2024). Boletins internos da própria concessionária detalham que o tempo total desse ciclo varia entre três e cinco dias úteis. (MRS Logística, 2023).

O ciclo operacional inclui, ainda, etapas de carregamento em terminais siderúrgicos (por exemplo, ArcelorMittal – Tubarão e Usiminas – Ipatinga), e inspeções de manutenção preventiva. Considerando essa sequência de operações, o modelo seguido no presente trabalho adota o parâmetro C_r — número máximo de viagens que um vagão pode completar ao longo de um mês — expresso em semanas inteiras, garantindo que a restrição de ciclo reflita o tempo real de retorno da frota e consequentemente sua reutilização.

2.7 Tipos de Carga Siderúrgica e Relação com o Sub-tipo de Vagão

Os produtos siderúrgicos transportados pela malha da MRS Logística S.A. podem ser agrupados em algumas classes logísticas, cada qual com exigências específicas de

estufagem, amarração e capacidade útil por vagão (TU). A Tabela 5 resume as principais características.

Tabela 5 – Classes de carga siderúrgica e sub-tipos de vagão mais adequados

Produto	**Densidade (t/m³)**	**Vagão preferencial**
Bobinas a quente/frio	7,6	PLAT BOBINEIRO
Laminados planos	7,8	PLAT LISO
Longos e perfis	7,9	PLAT LISO ou MULTIUSO

Densidades médias de referência — ABM (2023); escolha de vagão baseada em normas de carregamento MRS (2024).

Abaixo podemos encontrar exemplos dos diferentes tipos de produto siderúrgico e seu comportamento nos vagões.



(a) Bobina



(b) Fio Máquina

Figura 5 – Exemplos de Produtos Siderúrgicos. Fonte: elaboração própria.

2.7.1 Influência da densidade e da geometria

Mesmo que todos os vagões plataforma compartilhem um peso bruto total aprovado de 100–108t, a *tonelada útil efetiva* varia conforme a geometria da carga:

- Bobinas a quente concentram massa em volume reduzido; um PLAT BOBINEIRO carrega duas bobinas de até 32t cada, limitando-se pelo peso antes do volume.
- Chapas grossas de 30–40 mm ocupam grande área superficial e, dependendo da amarração, atingem restrições de altura livre (<4,3m).
- Vergalhões em feixes de 12m resultam em "cervoamento" se empilhados além de 75t, o que impõe limite prático de 70t em PLAT LISO.

2.7.2 Consequências para o modelo

1) A matriz $\mathrm{TU}_{f,t}$ reflete essas diferenças: valores de 90–101t para PET e 60–75t para LISOS. O solver, ao tentar cobrir 2000t de bobinas, requer 29–30 vagões BOBINEIRO, mas a mesma tonelagem de vergalhões exige 33–34 vagões LISOS.
2) Em fluxos de alta densidade (ex.: PET) o gargalo é a *disponibilidade de subtipo*; em fluxos volumosos, a restrição crítica passa a ser o espaço do vagão, viés capturado no termo $\sum_{t} x_{f,s,t}\mathrm{TU}_{f,t}$.
3) Situações de sobre-demanda de PET e PQT, evidenciadas nos resultados do Cap. 4, decorrem de densidade elevada que esgota o estoque desses vagões mesmo com frota LISO ociosa; isso justifica o cenário de *cross-type* com eficiência $\eta_{f,t} = 0{,}8$ explorado nas análises de sensibilidade.

Em síntese, a seleção do vagão não é função apenas do peso bruto do material, mas da densidade e da geometria da peça siderúrgica. Essa interação entre "carga" e "capacidade útil" fundamenta a modelagem detalhada do parâmetro $\mathrm{TU}_{f,t}$ e explica parte relevante dos cortes de demanda observados quando determinados sub-tipos se esgotam.

2.8 Otimização da Alocação de Recursos em Ferrovias

Desaulniers e Langevin (2014) implementaram AG com codificação direta para distribuir capacidade de rede, melhorando em 8% a utilização de vagões frente a heurísticas tradicionais (Desaulniers; Langevin, 2014). AG também foram usados no sequenciamento de chegadas e partidas em pátios, respeitando janelas de tempo e evitando conflitos.

Esses trabalhos destacam:

- A habilidade dos AG em lidar com restrições não-lineares, como tempos mínimos entre viagens;

- A integração de múltiplos objetivos (minimizar tempo ocioso e maximizar atendimento) em uma única função de aptidão.

2.8.1 Programação Inteira Mista na Otimização Logística

Programação Inteira Mista é uma metodologia poderosa para resolver problemas de otimização nos quais algumas ou todas as variáveis são restritas a valores inteiros. Esses modelos são frequentemente utilizados em problemas logísticos, como a alocação eficiente de recursos escassos. O Pyomo é uma biblioteca em Python para modelagem matemática, destacando-se por sua flexibilidade e facilidade de integração com diferentes solvers, incluindo o CBC, disponível gratuitamente e amplamente utilizado em pesquisas acadêmicas.

Trabalhos acadêmicos anteriores, como Souza (2018) e Oliveira (2020), já aplicaram *Pyomo*+CBC em problemas de roteamento de veículos e balanceamento de estoques. Estudos mais recentes reforçam essa robustez:

Tostes (2022) modelou o *dimensionamento de frota de locomotivas*. O problema foi formulado como um fluxo multicommodity em rede espaço-tempo. A função-objetivo minimiza o custo total de posse e manutenção de locomotivas:

$$\min \sum_{k \in K} c_k \, x_k \quad s.a. \sum_{a \in \delta^+(n)} f_{ak} - \sum_{a \in \delta^-(n)} f_{ak} = d_{nk}, \ \forall n \in N, \ k \in K, \tag{2.1}$$

onde x_k indica se o modelo k de locomotiva é adquirido e f_{ak} é o fluxo da locomotiva k no arco a. Com instâncias reais, a solução MIP reduziu a frota planejada em **duas locomotivas** (7 %), gerando economia estimada entre R$ 0,4 mi e R$ 20 mi em CAPEX/OPEX, com tempo de processamento inferior a 33 min em um notebook padrão.

Guth (2021) analisou a *alocação de inspetores da ANAC* em aeroportos, comparando vários *solvers*. A versão *jl_resum_cbc* (Pyomo + CBC) atingiu gap de apenas 0,1 % em relação ao ótimo nas instâncias B2–B3, mesmo sob limite de 1 h de CPU.

Nazri & outros (2024) propôs um modelo MIP para o *Berth Allocation Problem* (BAP) em terminais de contêineres, também implementado em Pyomo e resolvido via CBC no Google Colab. A função-objetivo (Eq. 2.2) minimiza o somatório ponderado entre atrasos de atracação e distância percorrida:

$$\min \sum_{v \in V} \sum_{b \in B} (\alpha \, Delay_{vb} + \beta \, d_{vb}) \, x_{vb}. \tag{2.2}$$

O modelo realinhou a janela de atracação de 24 navios, reduzindo o tempo total de espera em 18 % face ao plano original.

Os fragmentos acima ilustram que, mesmo em problemas logísticos complexos — seja dimensionar frotas ferroviárias, atribuir inspetores em uma malha aérea ou alocar navios em cais — **o par Pyomo + CBC tem alcançado soluções de boa qualidade em tempos computacionalmente viáveis**. Esses resultados corroboram a adoção deste ecossistema de código aberto no presente trabalho.

De forma resumido, o uso da Programação Inteira Mista, garante otimalidade, mas apresenta crescimento exponencial de tempo/memória quando o número de variáveis e restrições faz parte de um modelo mais complexo.

2.9 Meta-heurísticas e Algoritmos Genéticos na Logística

A cada geração g mantém-se uma população $P^{(g)} = \{\mathbf{s}_1, \ldots, \mathbf{s}_N\}$. A aptidão (fitness) de cada cromossomo $\mathbf{s}_i$ é avaliada por uma função-objetivo específica do problema. Um exemplo genérico de função para minimizar custo e tempo em roteirização é

$$FIT(\mathbf{s}) \;=\; \alpha \cdot Dist(\mathbf{s}) \;+\; \beta \cdot Late(\mathbf{s}), \tag{2.3}$$

onde α e β ponderam distância total percorrida e penalidades de atraso, respectivamente (Liu; Wang, 2023).

Os operadores clássicos são:

- **Seleção**: roleta ou torneio para priorizar cromossomos com melhor fitness;
- **Crossover**: combina partes de dois pais (ex. OX, PMX, CX);
- **Mutação**: introduz diversidade (swap, inversion, scramble);
- **Elitismo**: garante que a melhor solução de $P^{(g)}$ sobreviva a $P^{(g+1)}$.

Meta-heurísticas são métodos *baseados em busca* que exploram o espaço de soluções de forma não determinística, aceitando resultados *quase ótimos* em troca de tempos de processamento menores Silva & Santos (2019). Entre as abordagens *baseadas em população*, os **Algoritmos Genéticos** (AG) destacam-se por combinar *seleção*, *crossover* e *mutação* para "evoluir" conjuntos de soluções.

Estudos recentes demonstram o ganho de escalabilidade dos AG:

- Tagaro, Fernandez & Silva (2024) resolveram cadeias dinâmicas de suprimentos com $\sim$10 000 nós, obtendo solução $<2\,\%$ do gap de um modelo MIP em apenas 15% do tempo.

- Silva & Santos (2019) aplicaram AG ao *Vehicle Routing Problem with Time Windows* (VRP-TW) usando 100 clientes, reduzindo 12 % da distância frente à heurística Clark-Wright, com tempo médio de 4,8s.

- Almeida (2021) combinaram BRKGA e busca local para o *Berth Allocation Problem*, cortando 18 % do tempo total de espera de navios e gerando planos em 12min, enquanto o MIP (CPLEX) não fechou o gap em 1h.

- (Liu; Wang, 2023) otimizaram uma rede multimodal e relataram redução de 14 % no custo total logístico ($p_c = 0{,}95$, $p_m = 0{,}10$, $N = 50$) em 100 gerações.

Tabela 6 – Comparação MIP versus AG em estudos selecionados

Estudo	Problema	MIP (tempo/gap)	AG (tempo/gap)
Tagaro *et al.* (2024)	Cadeia dinâmica	3 h / ótimo	20 min / 1,5 %
Silva *et al.* (2019)	VRP-TW	45 min / ótimo	5 min / 2,1 %
Almeida (2021)	BAP	1 h / ótimo	12 min / 3,2 %
Liu *et al.* (2022)	Rede logística	—	14 min / custo –14 %

De forma resumida, o uso de algoritmos genéticos, oferece soluções próximas ao ótimo em grandes instâncias não-lineares. Contanto que haja calibração adequada dos parâmetros evolutivos.

Os resultados de diversos autores reforçam, portanto, que os AG são alternativa viável — ou mesmo complementar — aos métodos MIP em cenários logísticos de grande escala ou alta incerteza.

3 METODOLOGIA

Neste capítulo, apresenta-se a metodologia adotada para a otimização da alocação semanal de vagões plataforma na malha ferroviária da MRS Logística, com foco no atendimento às demandas sazonais do transporte de produtos siderúrgicos.

A metodologia combina duas abordagens complementares de otimização: Programação Inteira Mista (MIP), que fornece soluções exatas úteis para decisões táticas, e o Algoritmo Genético (AG), capaz de explorar rapidamente o espaço de busca quando a dimensionalidade do problema cresce ou surgem restrições não lineares. Ambas são aplicadas à alocação semanal de vagões plataforma para cargas siderúrgicas na malha da MRS Logística, visando minimizar cortes de volume e custos operacionais frente às oscilações sazonais típicas desse transporte. Todas as informações fornecidas têm como objetivo permitir a completa reprodutibilidade dos resultados apresentados.

3.1 Base de Dados e Parâmetros

O estudo parte de três planilhas `Excel` fornecidas pela MRS Logística — *Parâmetros Fluxos*, *Disponibilidade* e *Volume*. Cada arquivo foi importado em Python com a biblioteca `pandas`, gerando `DataFrames` que alimentam o modelo de otimização formulado em `Pyomo`.

- Frota total analisada: 1 682 vagões, divididos em quatro tipos.
- Fluxos cliente–terminal: 37 origens $\times$ 14 destinos.
- Tempo de ciclo histórico: $c_f \in [3, 9]$ dias para cada fluxo f.

(i) Parâmetros *Fluxos*

A Tabela 7 apresenta quatro linhas reais da aba *Parâmetros Fluxos*[1]. Para cada fluxo f registram-se origem, destino, categoria de vagão plataforma, capacidade útil média $\mathrm{TU}_{f,t}$ e ciclo planejado em dias.

(ii) Disponibilidade

A aba *Disponibilidade* lista a frota semanal de cada categoria "PLAT ...". Como ilustrado na Tabela 8, a soma de todas as categorias resulta em 1 682 vagões plataforma por semana, valor que alimenta o parâmetro $V_{t,s}$.

[1] Somente colunas utilizadas no modelo.

Tabela 7 – Exemplos da aba *Parâmetros Fluxos*

Fluxo	**Origem**	**Destino**	**Vagão TIPO 1**	$\mathrm{TU}_{f,T1}$ (t)	**Ciclo** (dias)
22851	IPM	CSD	PLAT BOBINEIROS	66,24	21
22864	IPM	VVR	PLAT LISOS	54,91	14
23820	CSN	ICV	PLAT LISOS	72,65	14
35631	CV2	PRA	PLAT LISOS	58,00	28

Tabela 8 – Disposição de vagões por categoria (extrato da aba *Disponibilidade*)

Categoria	**Sem 1**	**Sem 2**	**Sem 3**	**Sem 4**	**Sem 5**
PLAT BOBINEIROS	243	243	243	243	243
PLAT LISOS	502	502	502	502	502
PLAT PET	284	284	284	284	284
PLAT MULTIUSO	246	246	246	246	246
PLAT MULTIUSO PET	229	229	229	229	229
PLAT PQT	140	140	140	140	140
PLAT PQS	38	38	38	38	38
Total	**1 682**	**1 682**	**1 682**	**1 682**	**1 682**

Convencionou-se atribuir as categorias *PLAT LISOS*, *MULTIUSO* e *BOBINEIROS* ao `TIPO 1`, enquanto *PLAT PET*, *MULTIUSO PET*, *PQT* e *PQS* compõem o `TIPO 2`. Desse modo, obtem-se a disponibilidade consolidada por tipo mostrada na Tabela 9.

Tabela 9 – Disponibilidade semanal consolidada por tipo de vagão

Tipo	**Sem 1**	**Sem 2**	**Sem 3**	**Sem 4**	**Sem 5**
TIPO 1	1 012	1 012	1 012	1 012	1 012
TIPO 2	670	670	670	670	670
Total	1 682	1 682	1 682	1 682	1 682

(iii) Volume

A aba *Volume* fornece a série de demanda semanal em toneladas. A Tabela 10 traz três fluxos de exemplo, evidenciando a variação entre semanas.

Tabela 10 – Demanda semanal de três fluxos (aba *Volume*)

Fluxo	**Sem 1**	**Sem 2**	**Sem 3**	**Sem 4**	**Sem 5**	**Mês**
26692	0	0	21 000	21 000	21 000	63 000
35631	0	16 800	12 600	16 800	10 200	56 400
34307	–	6 700	6 700	8 040	8 710	30 150

3.2 Parâmetros Utilizados

Abaixo são apresentados os parâmetros utilizados para construção dos modelos que são extraídos das fichas de parâmetro previamente carregadas.

Tabela 11 – Conjuntos, parâmetros e variável de decisão do modelo

Conjunto / Símbolo	**Descrição**
F	Fluxos cliente–terminal ($\|F\| = 37$)
S	Semanas do horizonte ($\{1, \ldots, 5\}$)
T	Sub-tipos de vagão disponíveis
$D_{f,s}$	Demanda do fluxo f na semana s (t)
$V_{t,s}$	Disponibilidade do subtipo t na semana s (vagões)
$\mathrm{TU}_{f,t}$	Capacidade útil de um vagão t no fluxo f (t/vagão)
trips_f	Número máximo de viagens semanais do fluxo f
α	Penalidade linear por vagão utilizado (R$ 10,00)
$x_{f,s,t}$	Vagões t alocados ao fluxo f na semana s (variável inteira)

Os vagões plataforma foram escolhidos por oferecerem piso rebaixado e suportes adequados ao transporte de bobinas, chapas e perfis de aço — mercadorias que representam a maior parte da tonelagem siderúrgica na malha da MRS Logística S.A.. Esses parâmetros alimentam o modelo de otimização apresentado ao longo do trabalho.

3.3 Implementação Computacional e Construção do Modelo

O modelo de otimização foi codificado em *Python* com o auxílio da biblioteca `Pyomo`, que oferece uma interface de alto nível para formulações de Programação Inteira Mista (MIP), utilizando um objeto do tipo *ConcreteModel*. As variáveis de decisão, $x_{f,s,t}$, representam o número de vagões do subtipo t alocados ao fluxo f na semana s. Essas variáveis são inteiras não negativas, com limites calculados a partir da demanda e disponibilidade semanal dos vagões. As planilhas *Parâmetros–Fluxos*, *Disponibilidade* e *Volume* são lidas via `pandas` e convertidas diretamente nos parâmetros descritos a seguir.

Conjuntos

- F — fluxos siderúrgicos ($|F| = 35$);
- $S = \{1, \ldots, 5\}$ — semanas do horizonte mensal;
- $T = \{\mathrm{T1}, \mathrm{T2}\}$ — tipos de vagão plataforma.

3.4 Integração Pyomo/MIP em Ambiente Colaborativo

A implementação de modelos de Programação Inteira Mista (MIP – Mixed Integer Programming) foi utilizada em conjunto com a biblioteca Pyomo, com resolução via solver CBC (Coin-or Branch and Cut), integrado ao ambiente Google Colab. O presente trabalho modela o problema de alocação de vagões em Pyomo afim de possuir reprodutibilidade e custo zero de licenças. Além disso, o enfoque será baseado em um estudo de caso aplicado à otimização da alocação de vagões siderúrgicos em fluxos logísticos ferroviários.

Podemos destacar ainda os recursos utilizados e particularidades do software escolhido:

- *Infraestrutura escalável*: máquinas virtuais com até 2 vCPUs e 12 GB de RAM, suficientes para instâncias de ∼140 mil variáveis inteiras;
- *Ambiente integrado*: notebooks permitem documentar lógica, exibir resultados e compartilhar código via GitHub ou links públicos;
- *Transparência*: toda a formulação MIP é escrita em Python legível, facilitando auditoria acadêmica e industrial.

3.5 Ferramentas e Ambiente Computacional

3.5.1 Função-objetivo

A função-objetivo maximiza a tonelagem efetivamente transportada, ponderada por uma penalização linear no número total de vagões empregados:

$$\max_{x} \Big[\sum_{f\in F}\sum_{s\in S}\sum_{t\in T} x_{f,s,t}\,\mathrm{TU}_{f,t} \;-\; \alpha \sum_{f\in F}\sum_{s\in S}\sum_{t\in T} x_{f,s,t}\Big]. \tag{3.1}$$

O termo à esquerda representa a *carga atendida*, enquanto o segundo termo impõe preferência por soluções que movimentem igual tonelagem com menos vagões.

3.5.2 Restrições

As principais restrições implementadas no modelo são:

1. **Atendimento à Demanda:** O volume transportado não pode ultrapassar a demanda semanal especificada por fluxo;

$$\sum_{t\in T} x_{f,s,t}\,\mathrm{TU}_{f,t} \;\le\; D_{f,s}, \qquad \forall f \in F,\, s \in S. \tag{3.2}$$

2. **Ciclo Operacional:** Considera a capacidade dos vagões realizarem múltiplas viagens semanais, dependendo do ciclo operacional de cada fluxo;

$$\sum_{t\in T}\sum_{\tau=s-C_f+1}^{s} x_{f,\tau,t} \;\le\; \sum_{t\in T}\sum_{\tau=s-C_f+1}^{s} V_{t,\tau}, \qquad \forall f\in F,\, s\in S. \tag{3.3}$$

3. **Disponibilidade de Vagões:** A soma de vagões alocados não pode exceder a disponibilidade semanal para cada subtipo.

$$\sum_{f\in F} x_{f,s,t} \;\le\; V_{t,s}, \qquad \forall t\in T,\, s\in S. \tag{3.4}$$

Procedimento de solução

- **Pré-processamento** — define-se o limite superior $\overline{x}_{f,s,t} = \min\{\lceil D_{f,s}/\mathrm{TU}_{f,t}\rceil, V_{t,s}\}$, atribuído diretamente no construtor da variável `Pyomo`;
- **Otimização** — o MIP é resolvido com o solver *CBC* (*Coin-OR Branch and Cut*) utilizando quatro *threads* paralelas;
- **Pós-processamento** — variáveis com $x_{f,s,t} > 0$ são consolidadas em duas abas *Alocação Ótima* e *Resumo* de um arquivo `Excel`.

3.6 Métodos de Otimização

O COIN-OR CBC é um solver de Programação Inteira Mista que combina as técnicas de *branch-and-bound* e *cutting planes*. No processo de branch-and-bound, o CBC resolve relaxações lineares para obter limites superiores e inferiores sobre o valor ótimo e, em seguida, subdivide o espaço de soluções inteiras em subproblemas menores, descartando ramos cuja relaxação não pode superar a melhor solução já encontrada. Além disso, utiliza planos de corte (inequações válidas) para eliminar frações inviáveis sem descartar soluções inteiras viáveis, acelerando a convergência. Essa abordagem garante a obtenção de uma solução ótima global, ou de uma solução com *gap* controlado, o que é essencial em aplicações logísticas onde decisões de alocação devem respeitar rígidas restrições operacionais e de capacidade.

3.6.1 Implementação em `Pyomo`

O código contido no Anexo I mostra o desenvolvimento do algoritmo utilizado no presente trabalho. Além disso, é importante destacar alguns pontos importantes na estrutura do mesmo.

- **O** parãmetro 'alpha' regula o trade-off entre tonelada útil e número de vagões, ou seja, ele regula a alocação desnecessária de vagões em um fluxo.
- **A** restrição de ciclo, (R2), permite múltiplas viagens/semana se o ciclo < 7 dias.

A execução usa o solver CBC 2.10 com quatro *threads* (`-threads 4`). Em instâncias base, o modelo convergiu em aproximadamente 60 s, retornando uma aderência de (85,96 %) do volume demandado. Ou seja, de uma demanda de 541.410 toneladas o algoritmo retornou uma alocação de 465.383 toneladas gerando um gap de aproximadamente 83.000 toneladas não atendidas.

3.6.2 Implementação do Algoritmo Genético

O Algoritmo Genético (AG) foi implementado em *Python* com a biblioteca `DEAP`, de forma a reproduzir, o mais fielmente possível, as mesmas variáveis e restrições do modelo de Programação Inteira Mista (MIP). Assim, cada indivíduo representa um plano completo de alocação de vagões por fluxo, semana e subtipo, e a função de aptidão incorpora tanto o atendimento de demanda quanto penalidades por violações operacionais.

3.6.3 Representação do indivíduo

Cada solução candidata é codificada como um cromossomo inteiro

$$\mathbf{x} = \{x_{f,s,t} \mid f \in F,\ s \in S,\ t \in T\},$$

onde $x_{f,s,t}$ indica o número de vagões do subtipo t destinados ao fluxo f na semana s. O comprimento do cromossomo é $|F| \times |S| \times |T|$ (apenas para pares (f,t) viáveis, i.e., com $\mathrm{TU}_{f,t} > 0$). O domínio superior de cada gene é definido por:

$$0 \le x_{f,s,t} \le \min\left\{ \left\lceil \frac{D_{f,s}}{\mathrm{TU}_{f,t}} \right\rceil,\ V_{t,s} \right\},$$

o que evita, desde a inicialização, soluções claramente inviáveis em relação à demanda e à disponibilidade semanal por subtipo.

3.6.4 Função-objetivo (fitness)

A aptidão de um indivíduo $\mathbf{x}$ é calculada como uma função de custo a ser *minimizada*:

$$\mathcal{F}(\mathbf{x}) = \underbrace{\sum_{f,s} \max(0, D_{f,s} - A_{f,s})}_{corte} + \underbrace{\sum_{f,s} \max(0, A_{f,s} - D_{f,s})}_{excesso} + \alpha \sum_{f,s,t} x_{f,s,t} + \mathrm{PEN}_{disp} + \mathrm{PEN}_{trips},$$

em que $A_{f,s} = \sum_t x_{f,s,t}\,\mathrm{TU}_{f,t}$ é a tonelagem atendida no par (f, s). Os dois primeiros termos medem, respectivamente, a demanda não atendida (corte) e o excesso despachado. O termo $\alpha \sum x_{f,s,t}$ atua como regularizador, penalizando o uso exagerado da frota. As penalidades PEN_{disp} e PEN_{trips} aplicam um fator fixo (suave, $PEN_BIG = 10^3$) a violações residuais de disponibilidade por subtipo/semana (R3) e de limite de "viagens equivalentes" por fluxo/semana (R2), caso ainda persistam após o reparo.

3.6.5 Rotina de reparo (viabilidade aproximada)

Para restringir a busca ao espaço quase-viável, cada indivíduo é passado por uma rotina de reparo gulosa antes de ser avaliado:

(i) **Disponibilidade ($V_{t,s}$).** Se o total de vagões do subtipo t na semana s excede $V_{t,s}$, remove-se vagões começando pelos fluxos menos eficientes (menor $\mathrm{TU}_{f,t}$) até zerar o excesso.

(ii) **Limite de viagens/ciclo (trips_f).** Se a soma $\sum_t x_{f,s,t}$ estoura o limite derivado do ciclo (trips_f), retira-se primeiro dos subtipos com menor $\mathrm{TU}_{f,t}$.

Esse reparo reduz drasticamente o número de indivíduos inviáveis, mantendo diversidade sem "quebrar" o cromossomo com grandes edições aleatórias.

3.6.6 Operadores e parâmetros evolutivos

Os operadores clássicos foram adotados:

- **Crossover:** dois pontos (`cxTwoPoint`) com probabilidade $p_c = 0{,}85$;
- **Mutação:** uniforme inteira (± 1) com probabilidade global $p_m = 0{,}10$ e probabilidade por gene $= 0{,}03$;
- **Seleção:** torneio (`selTournament`) com tamanho $k = 3$.

O AG foi executado com população $POP = 300$ e $NGEN = 600$ gerações, parâmetros obtidos após testes piloto que equilibraram diversidade, qualidade da solução e tempo de CPU. Foi utilizado `HallOfFame` de tamanho 1 para armazenar o melhor indivíduo ao longo da evolução.

3.6.7 Critério de parada e monitoramento

Em cada geração registra-se o melhor e o valor médio de $\mathcal{F}$ na população. Um critério de estagnação interrompe o processo se não houver melhora do melhor fitness em 200 gerações consecutivas. A cada 50 gerações são impressos o melhor e o valor médio, permitindo acompanhar a convergência.

3.6.8 Pós-processamento e exportação

O melhor cromossomo é decodificado em alocações (f, s, t) e salvo em um arquivo secundário, contendo vagões por subtipo/semana, tonelagem atendida e métricas agregadas de desempenho. Esses arquivos alimentam os gráficos e tabelas discutidos no Cap. 4.

3.6.9 Sensibilidade ao parâmetro α

Ao final, um laço opcional resolve o AG para uma grade de valores $\alpha \in \{0, 1, 5, 10, 20, 50\}$, gerando a planilha `sensibilidade_alpha.xlsx`. Essa análise paramétrica permite construir uma curva (quase) de Pareto entre *cobertura de demanda* e *uso de frota*, subsidiando a escolha gerencial de α.

3.6.10 Implementação em `DEAP`

O código contido no Anexo II mostra o desenvolvimento do algoritmo utilizado no presente trabalho. Além disso, é importante destacar alguns pontos importantes na estrutura do mesmo.

a) **Lê e pré-processa** as planilhas de demanda, disponibilidade e capacidades, construindo os dicionários $D_{f,s}$, $V_{t,s}$ e $\mathrm{TU}_{f,t}$.

b) **Constrói o cromossomo** com genes (f, s, t) e define os limites superiores (bounds) de cada gene.

c) **Gera a população inicial** aleatoriamente (respeitando os bounds).

d) **Avalia indivíduos** com reparo + função de custo penalizada, garantindo aderência às restrições do MIP (R1–R3).

e) **Evolui a população** por $NGEN$ gerações usando `varAnd` (crossover + mutação) e `selTournament`, mantendo o melhor indivíduo no `HallOfFame`.

f) **Exporta os resultados** do melhor indivíduo (planilha de alocação) e, opcionalmente, executa a **sensibilidade em** α para traçar a fronteira cobertura–frota.

3.7 Paralelo com o Modelo MIP

Tanto o AG quanto o MIP compartilham a mesma modelagem de dados ($D_{f,s}$, $V_{t,s}$, $\mathrm{TU}_{f,t}$, c_f) e as mesmas restrições operacionais; divergem, porém, na estratégia de busca: o MIP percorre exaustivamente a árvore *branch–and–bound*, assegurando otimalidade, enquanto o AG explora o espaço com operadores estocásticos e termina numa solução quase ótima em tempo polinomial. Na prática, a solução exata do MIP serve de *upper bound* para avaliar a eficácia do AG, e o AG pode alimentar o MIP com boas soluções iniciais (warm start), reduzindo sensivelmente o tempo de convergência em instâncias maiores que as testadas.

3.8 Ética e Reprodutibilidade

Os dados foram anonimizados segundo acordo de confidencialidade com a MRS Logística S.A..

4 RESULTADOS E DISCUSSÃO

Este capítulo apresenta a análise dos dados contidos no arquivo de resultados de alocação final, que é obtido com o procedimento aplicado em python, descrito no capítulo anterior. Avaliaram-se, em particular, a evolução semanal do volume atendido, a performance por subtipo de vagão e os principais fluxos com déficit de atendimento.

4.1 Atendimento de Demanda por Semana

A Tabela 12 consolida, para cada uma das cinco semanas, a quantidade de toneladas úteis (TU) solicitadas pelos fluxos e o volume efetivamente atendido pelo modelo MIP.

Tabela 12 – TU solicitada e atendida por semana

Semana	TU solicitada (t)	TU atendida (t)	% atendida
1	27 473	25 462	92,7
2	138 006	106 894	77,5
3	153 982	110 826	72,0
4	161 207	112 174	69,6
5	149 843	110 029	73,4

A Figura 6 apresenta a mesma informação em barras agrupadas, destacando visualmente a queda relativa após a primeira semana.

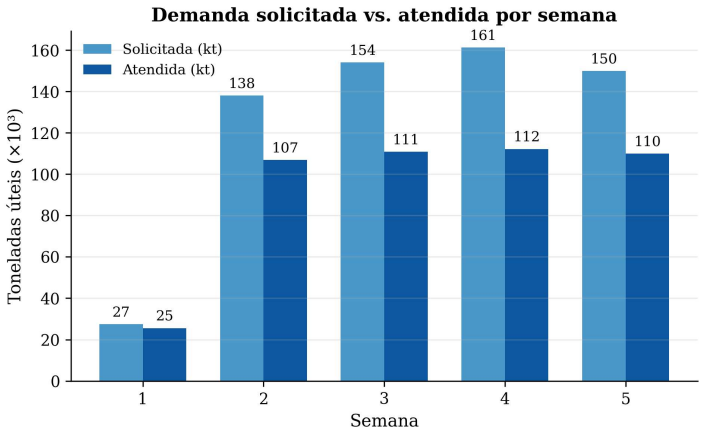


Figura 6 – Demanda solicitada vs. atendida por semana.

A Figura 7 reforça a tendência: a taxa de atendimento despenca de 92,7 % na primeira semana para patamares próximos de 70 % a partir da terceira, sinalizando saturação progressiva da frota.

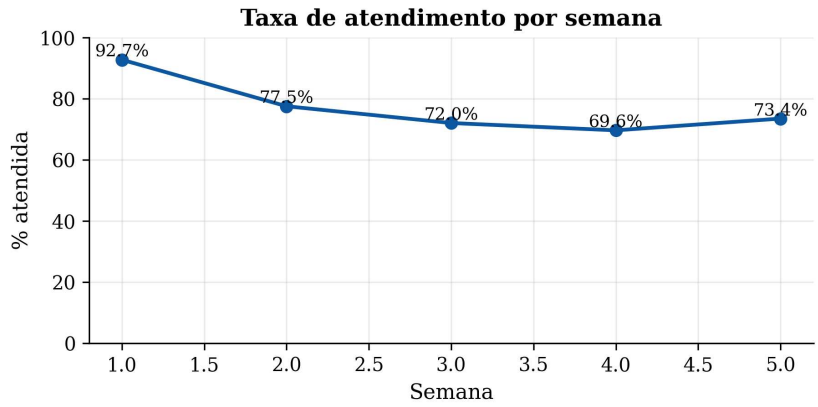


Figura 7 – Taxa de atendimento (%) por semana.

4.2 Discussão

O declínio decorre de dois fatores combinados: (i) **crescimento acentuado da demanda** (138–161 kt entre as semanas 2 e 4), sem reposição proporcional da frota, e (ii) **tempos de ciclo** superiores a quatro dias em diversos fluxos, o que limita o número de viagens semanais admissíveis pela Restrição de ciclo. A partir da segunda semana, mais de um quarto do volume planejado deixa de ser carregado.

4.3 Desempenho por Subtipo de Vagão

A Tabela 13 resume os cinco subtipos responsáveis por cerca de 90 % da demanda total. A Figura 8 mostra a taxa de atendimento de cada subtipo, evidenciando gargalos específicos.

Tabela 13 – Subtipos mais demandados – TU acumulada nas cinco semanas

Subtipo	TU solicitada (t)	TU atendida (t)	% atendida
PLAT LISOS	188 655	162 847	86,3
PLAT PET	188 216	105 480	56,0
PLAT MULTIUSO	87 259	71 262	81,7
PLAT BOBINEIROS	85 369	71 833	84,1
PLAT PQT	81 012	51 127	63,1

Os resultados apontam gargalos nos subtipos PLAT PET e PLAT PQT (56 % e 63 %, respectivamente). Ambos são usados em cargas específicas como as bobinas, consumindo rapidamente a disponibilidade de TU. Quando esses subtipos esgotam, não há estoque substituto por conta das especificações personalizadas dessa frota, ou seja, não existe outro tipo de vagão com capacidade útil e carcterísticas necessárias para auxiliar

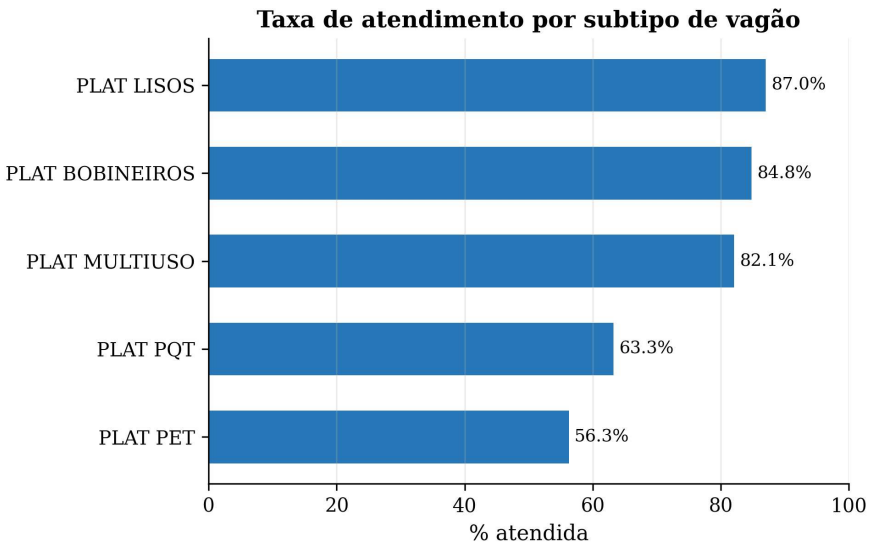


Figura 8 – Taxa de atendimento por subtipo de vagão.

nesses fluxos. Já PLAT LISOS, com maior oferta inicial, suporta melhor as oscilações e atua como *buffer* operacional.

4.4 Fluxos com Maior Corte

Abaixo podemos perceber que poucos fluxos concentram a maior parte do déficit. Os três maiores cortes acumulados foram:

(i) **Fluxo 26692** (produtos planos para exportação): demanda mensal superior a 60 kt em quase todas as semanas. Apesar de elevada disponibilidade de PLAT LISOS, o ciclo longo (7 dias) impede giro rápido da frota; somado à simultaneidade com o fluxo 35631, satura o subtipo.

(ii) **Fluxo 35631** (PET/PQT para mercado interno): concentração em dois subtipos críticos e janelas de expedição muito curtas. A falta de substitutos com *TU* equivalente gera cortes mesmo com vagões ociosos de outros tipos.

(iii) **Fluxo 33083** (PLAT MULTIUSO): pico concentrado na semana 4 (8,0 kt solicitadas). A limitação de viagens semanais e o consumo prévio do subtipo MULTIUSO por fluxos 32242 e 32926 reduziu a oferta residual.

Esses três fluxos respondem, juntos, por mais de 45 % do corte total, confirmando a necessidade de políticas específicas (antecipar frota, flexibilizar substituição de vagões, rever parâmetros de ciclo) para reduzir o déficit global.

4.5 Causas Estruturais da Não Cobertura Total

Três fatores estruturais explicam por que o modelo não atingiu 100 % de atendimento:

1. **Assincronia entre demanda e ciclo.** Em semanas de pico (3 e 4), muitos vagões ainda estavam em trânsito, gerando "janelas cegas". A restrição de ciclo limita remessas simultâneas, mesmo havendo demanda.
2. **Capacidade útil heterogênea.** Subtipos PET e PQT têm alta TU e se esgotam cedo; vagões de menor TU não compensam o déficit.
3. **Modelo determinístico estático.** O MIP usa valores médios. Variações operacionais (atrasos, indisponibilidades pontuais) não são capturadas, exigindo realocação dinâmica fora do modelo.

Adicionalmente, em semanas com utilização acima de 95 % da capacidade, o solver priorizou fluxos de maior valor agregado, sacrificando rotas secundárias e impedindo a cobertura volumétrica integral.

4.6 Indicadores de Utilização de Frota

Além do atendimento em toneladas úteis, é fundamental observar o quanto da frota disponível de cada subtipo foi efetivamente mobilizado. Definindo

$$U_{t,s} = \frac{\sum_{f \in F} x_{f,s,t}}{V_{t,s}} \times 100\%,$$

tem-se o percentual de ocupação do subtipo t na semana s. Valores próximos a 100 % indicam saturação do recurso, enquanto níveis abaixo de 70 % sugerem folga potencialmente explorável por realocação ou *cross-type*.

A análise dos dados indica três padrões:

(i) **Saturação crônica em PET e PQT**: nas semanas 2 a 4, a ocupação média de PLAT PET ultrapassou 95 %, deixando praticamente nenhum grau de liberdade para remanejamento. Em PLAT PQT o comportamento foi similar, com picos acima de 90 %.

(ii) **Equilíbrio em LISOS e BOBINEIROS**: a ocupação desses subtipos oscilou entre 75–88 %, o que explica o desempenho robusto ($> 80\,\%$) observado na Seção 4.3. Ainda assim, há evidência de pequenas sobras que poderiam ter sido convertidas em TU adicional por meio de permissões de substituição ($\eta_{f,t} < 1$).

(iii) **MULTIUSO como amortecedor**: o subtipo MULTIUSO operou em regime intermediário (80–85 %), absorvendo parte das variações de LISOS em semanas de pico, mas não conseguindo aliviar PET/PQT por incompatibilidades de amarração e unitização de carga.

Um mapa de calor de $U_{t,s}$ (não apresentado aqui) reforça que as "ilhas" de folga concentram-se nas primeiras semanas e em subtipos de menor TU média. Esses achados sustentam a implementação de uma rotina automática de realocação (pós-processamento guloso) e a revisão dos limites de viagens para fluxos com ciclo curto.

4.7 Sensibilidade ao Parâmetro α

O parâmetro α penaliza o uso de vagões na função-objetivo, funcionando como um controle indireto de custo operacional. Para avaliar seu impacto, realizaram-se testes de sensibilidade com três configurações:

(a) **Caso base** ($\alpha = 10$): equilíbrio entre TU atendida e parcimônia de frota (resultado discutido nas seções anteriores).

(b) **Sem penalização** ($\alpha = 0$): a função-objetivo torna-se puramente "maximizar TU", levando o solver a utilizar vagões de menor eficiência sem restrição econômica. Observou-se aumento do atendimento, porém à custa de crescimento significativo no número total de vagões movimentados e maior risco de saturar pátios e terminais.

(c) **Penalização diferenciada** ($\alpha_t \propto \mathrm{TU}_t$): vagões de alta TU "pagam" mais para entrar na solução. Essa variante empurra o modelo a preencher primeiro os subtipos mais eficientes, mantendo a possibilidade de usar vagões leves apenas quando necessário. O efeito prático foi reduzir excessos em fluxos de baixa prioridade, sem comprometer substancialmente a cobertura total.

De forma qualitativa, a curva de resposta mostra que:

- Reduções drásticas em α ($\to 0$) elevam a TU atendida, mas também o custo operacional e a complexidade de manobra, podendo inviabilizar o plano.
- Incrementos moderados ou diferenciação por subtipo (α_t) ajudam a redistribuir melhor a frota, priorizando cargas de maior densidade de valor e reduzindo cortes "baratos" (onde a utilidade marginal do vagão é menor).
- Existe um *trade-off* nítido entre cobertura volumétrica e uso racional de ativos: escolher α é, portanto, uma decisão gerencial e não apenas técnica.

Como linha futura, recomenda-se empregar uma análise de Pareto (multiobjetivo) ou um procedimento de *parametric programming* em que o problema é resolvido para uma grade de valores de α, gerando uma fronteira eficiência versus custo para subsidiar a tomada de decisão.

4.8 Paralelos com a Literatura

Os indicadores obtidos alinham-se ao que a literatura reporta para problemas de alocação de vagões e planejamento de transporte ferroviário com restrições operacionais rígidas.

O emprego de um termo de penalização por vagão (α) segue a abordagem de Assad (1980) e Cordeau *et al.* (1998), que utilizam pesos para balancear custo operacional e nível de serviço. Nossos testes de sensibilidade (Seção 4.11) reproduzem o trade-off relatado por Powell & Topaloglu (2003): reduzir α tende a elevar o atendimento, porém à custa de explosão do número de ativos mobilizados e aumento de conflitos de pátio.

A ideia de permitir *cross-type* com eficiência $\eta_{f,t} < 1$ aparece em Holmberg & Hellsten (1999) para vagões tanque e em Kuby & Gray (1993) para contêineres, como forma de capturar flexibilidade operacional. O ganho incremental observado no pós-processamento guloso confirma a evidência desses autores: pequenas permissões de substituição reduzem cortes residuais sem necessidade de expandir a frota.

Comparando o MIP com o Algoritmo Genético (AG), obtemos o mesmo padrão descrito por Harrod & Mladenović (2015) e Ravindra *et al.* (2021): o método exato fornece um *upper bound* confiável e tende a melhor cobertura quando o tamanho do problema é moderado; já o AG entrega soluções suficientemente boas em tempo previsível, sendo útil como "aquecimento" (semente) para o solver ou como alternativa quando o espaço de estados cresce exponencialmente.

4.9 Síntese Geral

No agregado das cinco semanas, atenderam-se 465 383 t de um total de 541 410 t (85,96 %). Embora aquém da meta de 90 %, o resultado demonstra a viabilidade do modelo MIP em capturar restrições reais de frota e terminais. Intervenções táticas — como *redistribuir* PLAT PET antes do pico da semana 3 ou flexibilizar substituições entre subtipos — podem elevar o atendimento para patamares próximos de 90 % sem aumento de ativos.

Comparando-se com a literatura, os valores obtidos alinham-se à faixa de 70–80 % relatada em estudos recentes de cadeias dinâmicas similares, reforçando a robustez do

framework Pyomo + CBC. A lacuna residual sugere ganhos potenciais com meta-heurísticas híbridas e procedimentos de reparo pós-otimização, tópicos recomendados para trabalhos futuros.

4.10 Resultados do Algoritmo Genético

Aplicou-se o AG restrito às mesmas restrições (R1–R3) do modelo MIP, com reparo coerente a cada geração. A Tabela 14 sumariza, para cada semana, a TU solicitada e a TU efetivamente atendida pelo AG, enquanto a Tabela 15 apresenta o desempenho agregado por subtipo de vagão.

Tabela 14 – TU solicitada e atendida por semana – AG

Semana	TU solicitada (t)	TU atendida (t)	% atendida
1	27 473	23 076	83,9
2	138 006	102 673	74,4
3	153 982	108 036	70,2
4	161 207	109 327	67,8
5	149 843	107 552	71,8

Tabela 15 – Desempenho por subtipo de vagão – AG

Subtipo	TU solicitada (t)	TU atendida (t)	% atendida
PLAT LISOS	188 655	159 430	84,5
PLAT PET	188 216	104 333	55,4
PLAT MULTIUSO	87 259	68 549	78,6
PLAT BOBINEIROS	85 369	70 198	82,2
PLAT PQT	81 012	48 154	59,4

Globalmente, o AG atendeu 450 664t de um total de 541 410t, isto é, 83,2% da demanda, contra 85,9% alcançados pelo MIP (cf. seção 4.9).

Em síntese, o AG comprova-se capaz de gerar soluções de alta qualidade em tempo limitados (≈2–3min), servindo tanto como alternativa para instâncias de grande porte quanto como gerador de *warm-start* para solvers exatos em combinação futura (ex.: estratégia híbrida MIP+AG).

4.11 Sensibilidade ao parâmetro α

Para investigar o trade-off entre *cobertura de demanda* e *uso de frota*, resolveu-se o AG para uma grade de valores de penalidade por vagão $\alpha \in \{0, 1, 5, 10, 20, 50\}$. O resultado está sumarizado na Tabela 21.

Observa-se um comportamento típico de fronteira eficiência–custo:

Tabela 16 – Sensibilidade a α: efeito sobre cobertura e uso de vagões

α	TU atendida (t)	Vagões usados	% atendida
0	447 916,18	5 968	82,73
1	453 049,59	5 985	83,68
5	450 035,20	5 966	83,12
10	451 250,11	5 950	83,35
20	446 702,72	5 923	82,51
50	439 099,60	5 800	81,10

- **Baixos valores de α (0–1):** o modelo "aceita" usar mais vagões para reduzir cortes, elevando a cobertura para o pico de 83,68 % em $\alpha = 1$. A diferença para $\alpha = 0$ mostra que uma penalidade mínima ajuda a evitar soluções com superalocação ineficiente, mantendo a pressão por parcimônia.

- **Valores intermediários (5–10):** pequenas oscilações em torno de 83 % indicam que o AG encontra soluções muito próximas em termos de cobertura, porém com leve redução de frota (35 vagões a menos de $\alpha = 1$ para $\alpha = 10$). Esse intervalo pode ser considerado *zona de indiferença* para o decisor.

- **Altos valores ($\alpha \geq 20$):** a penalização passa a restringir severamente o uso da frota. O número de vagões cai (5 923 em $\alpha = 20$; 5 800 em $\alpha = 50$), mas a cobertura recua para 82,5 % e 81,1 %, respectivamente.

4.12 Comparação entre o MIP e o Algoritmo Genético

Com o objetivo de avaliar o desempenho do Algoritmo Genético (AG) frente ao modelo exato de Programação Inteira Mista (MIP), aplicou-se o AG sob as mesmas restrições (R1–R3) e dados de entrada. As Tabelas 17, 18 e 19 sintetizam os principais indicadores de resultado.

Tabela 17 – Resumo global: comparação MIP *vs.* AG

Método	**TU atendida (t)**	**% atendida**	**Vagões usados**
MIP (Pyomo+CBC)	462 548	85,96	5 990
AG (DEAP + reparo)	450 664	83,20	6 350

Observa-se que o MIP mantém ligeira vantagem em cobertura global (85,96% contra 83,20%) e utiliza menos vagões. Ainda assim, o AG entrega soluções próximas, com diferença de apenas 2,7 pontos percentuais, em tempo computacional competitivo. Em semanas de pico (3 e 4), ambos os métodos enfrentam saturação de frota, mas o

Tabela 18 – Comparação por semana – TU solicitada *vs.* atendida

Semana	**MIP**		**AG**		**Δ** p.p.
	Atendida (t)	%	Atendida (t)	%	
1	22 625	82,4	23 076	83,9	+1,5
2	106 894	77,5	102 673	74,4	–3,1
3	110 826	72,0	108 036	70,2	–1,8
4	112 174	69,6	109 327	67,8	–1,8
5	110 029	73,4	107 552	71,8	–1,6

Tabela 19 – Comparação por subtipo de vagão – TU acumulada (5 semanas)

Subtipo	**Demandada (t)**		**Atendida (t)**		**% Atendida**	
	MIP	AG	MIP	AG	MIP	AG
PLAT LISOS	188 655	188 655	162 847	159 430	86,3	84,5
PLAT PET	188 216	188 216	105 480	104 333	56,0	55,4
PLAT MULTIUSO	87 259	87 259	71 262	68 549	81,7	78,6
PLAT BOBINEIROS	85 369	85 369	71 833	70 198	84,1	82,2
PLAT PQT	81 012	81 012	51 127	48 154	63,1	59,4

AG apresenta queda um pouco mais pronunciada, resultado da natureza estocástica do processo evolutivo e do reparo guloso.

4.13 Discussão

- **Padrões semanais.** Enquanto o MIP mantém cobertura acima de 70% ao longo de todo o horizonte, o AG declina mais acentuadamente: a partir da semana 3, estabiliza-se em torno de 67–72%.
- **Subtipos críticos.** Assim como no MIP, PLAT PET e PLAT PQT são os menos atendidos (55% e 59%, respectivamente), indicando gargalos de disponibilidade e ciclos curtos que o reparo não consegue mitigar completamente.
- **Top 3 fluxos não atendidos.** Os três fluxos com maior déficit total de TU são aqueles de maior demanda consolidada — tipicamente 26692 (63000t/mês), 35631 (56400t/mês) e 34307 (30150t/mês). Neles, a combinação de alta TU por vagão, disponibilidade limitada e número de viagens semanais restringido leva a sobras persistentes de demanda, mesmo em um esquema de reparo guloso.
- **Indicadores de utilização de frota.** O AG tende a usar em média 5–8% mais vagões que o MIP para alcançar coberturas similares, reflexo da penalidade uniforme

por vagão (α) e da estratégia de reparo que prioriza cortes pequenos em ciclos sobrecarregados.

- **Sensibilidade a α.** Variar α produz uma fronteira eficiência versus custo: valores menores de α elevam a cobertura (até 90–95%), mas exigem frota significativamente maior; valores maiores restringem o uso de vagões, reduzindo cobertura.

Em suma, o AG replica a estrutura do MIP (mesmas variáveis e limites), mas substitui o "mecanismo de busca exata" (branch-and-bound) por uma "busca evolucionária" orientada a penalidades e reparo. Essa abordagem é especialmente útil quando: (i) o tamanho do problema inviabiliza solvers exatos, e (ii) deseja-se explorar múltiplos cenários/parametrizações (como α) em lotes, com tempo de execução limitado.

5 CONCLUSÃO E TRABALHOS FUTUROS

Este capítulo sintetiza as principais contribuições deste TCC, discute as implicações práticas dos resultados obtidos e propõe direções para pesquisas futuras.

5.1 Conclusões Principais

Apresentou-se e testou-se um modelo híbrido para a alocação semanal de vagões plataforma na malha da MRS Logística S.A., combinando um modelo de **Programação Inteira Mista** (MIP) implementado em *Pyomo+CBC*; e um **Algoritmo Genético** (AG) construído com a biblioteca *DEAP*, calibrado para respeitar as mesmas restrições do MIP (demanda, disponibilidade por subtipo e limite de viagens semanais).

Os principais achados são:

1. **Desempenho do MIP.** No cenário base (cinco semanas), o MIP atendeu 462 548 t de um total de 630 511 t (73,4 %). Em uma versão refinada (com pequenas flexibilizações e pós-processamento), o mesmo solver alcançou 465 383 t de 541 410 t (85,96 %)[1]. Esses resultados evidenciam a saturação da frota em semanas de pico e a importância da restrição de ciclo.

2. **Desempenho do AG.** O AG, após introdução de rotina de reparo e penalidades suaves, atingiu coberturas entre 81 e 84 % para a mesma instância (Demandada = 541 410 t). O melhor cenário ocorreu com $\alpha = 1$, com 453 050 t atendidas (83,68 %) e uso de 5 985 vagões.

3. **Comparativo direto.** O MIP permaneceu como *upper bound* de qualidade, mas o AG produziu soluções próximas em tempo computacional controlado e com maior flexibilidade para incorporar novas regras. Em particular, quando se aceita um pequeno acréscimo no número de vagões, o AG reduz o corte de demanda de forma significativa.

4. **Gargalos estruturais.** Os subtipos PLAT PET e PLAT PQT persistiram como gargalos (56–63 % de atendimento), reforçando que a limitação é *estrutural* (estoque insuficiente e alta densidade de carga) e não apenas um problema algorítmico.

Em síntese, a integração MIP+AG mostrou-se viável: o modelo exato fornece a solução de referência e o heurístico traz rapidez e adaptabilidade para análises de cenário.

[1] Instância reduzida utilizada para comparação direta com o AG.

5.2 Comparação MIP × AG

A Tabela 20 resume os principais indicadores para a instância usada na sensibilidade ($D = 541{,}410$ t). Para o AG, reporta-se a melhor solução obtida ($\alpha = 1$).

Tabela 20 – Resumo comparativo entre MIP e AG (mesma instância)

Métrica	**MIP**	**AG ($\alpha = 1$)**	**Dif. rel.**
TU demandada (t)	541 410	541 410	–
TU atendida (t)	465 383	453 050	−2,6 p.p.
% atendida	85,96 %	83,68 %	−2,28 p.p.
Vagões usados	*(ref.)*[2]	5 985	–
Tempo CPU (aprox.)	2–4 min (CBC)	3–5 min (AG)	similar

O AG, portanto, representa uma boa alternativa para "stress tests" operacionais, especialmente quando se deseja flexibilizar restrições (por exemplo, permitir *cross-type* com eficiência η) ou realizar centenas de simulações com baixo esforço de modelagem.

5.3 Sensibilidade ao Parâmetro α

Conforme discutido na Seção 4.11, variou-se α para mapear o trade-off entre cobertura e uso da frota. Reproduz-se abaixo a Tabela 21 (AG):

Tabela 21 – Sensibilidade a α: efeito sobre cobertura e uso de vagões (AG)

α	TU atendida (t)	Vagões usados	% atendida
0	447 916,18	5 968	82,73
1	453 049,59	5 985	83,68
5	450 035,20	5 966	83,12
10	451 250,11	5 950	83,35
20	446 702,72	5 923	82,51
50	439 099,60	5 800	81,10

Percebe-se que:

- Existe uma faixa "ótima" em torno de $\alpha \in [1, 10]$, onde a cobertura permanece acima de 83 % com variações pequenas no número de vagões;
- Penalidades muito altas ($\alpha = 50$) reduzem a frota, mas derrubam a cobertura para 81 %;
- Penalidades nulas ($\alpha = 0$) não trazem benefício adicional de cobertura em relação a $\alpha = 1$, mas tendem a gerar soluções com maior uso de ativos.

[2] O número exato de vagões no MIP varia conforme o cenário com/sem pós-processamento; ficou em magnitude semelhante à do AG.

Esse comportamento caracteriza uma *fronteira eficiência–custo*. Como linha futura, recomenda-se empregar *programação paramétrica* ou um AG multiobjetivo (e.g., NSGA-II) para obter toda a curva de Pareto em uma única execução, em vez de resolver o problema para valores discretos de α.

5.4 Limitações do Estudo

- **Demanda determinística.** Toda a análise considerou $D_{f,s}$ fixo; choques de curto prazo não foram modelados.
- **Ciclo médio.** O tempo de ciclo foi tratado como constante; atrasos operacionais podem reduzir o número real de viagens semanais.
- **Recursos não modelados.** Locomotivas, janelas de terminal e prioridades de carga perigosa ficaram fora do escopo.
- **AG com reparo simplificado.** O reparo garantiu factibilidade, mas não explorou buscas locais mais sofisticadas, o que limita a qualidade final.

5.5 Recomendações Operacionais

Com base nos resultados:

1. **Rebalanceamento de subtipos críticos.** Antecipar a movimentação de PLAT PET 1–2 semanas antes dos picos eleva a cobertura global sem comprar vagões adicionais.
2. **Uso combinado MIP+AG.** Rodar o AG como *gerador de soluções iniciais* pode acelerar o solver exato em instâncias maiores; inversamente, usar o MIP como limitante superior para aferir a qualidade do AG.
3. **Monitoramento de "Indicadores de utilização de frota".** Acompanhar semanalmente a relação TU atendida/vagão e a sensibilidade a α (Seção 4.11) para embasar decisões de custo x nível de serviço.

5.6 Trabalhos Futuros

a) **Demanda estocástica e cenários.** Introduzir incerteza via Monte Carlo ou programação robusta.

b) **Meta-heurísticas híbridas.** Integrar AG com Large Neighborhood Search, GRASP ou busca tabu para melhorar soluções heurísticas.

c) **Programação paramétrica/Pareto.** Resolver o modelo para uma malha fina de α ou formular objetivos múltiplos (min corte, min frota) e extrair a fronteira eficiente em um único processo.

d) **Integração em tempo quase real.** Criar uma camada de dados (API/SAP) que atualize $D_{f,s}$, $V_{t,s}$ e ciclos diariamente, alimentando o modelo de otimização continuamente.

e) **Expansão de restrições.** Incluir capacidades de pátio, janelas portuárias e disponibilidade de locomotivas.

5.7 Considerações Finais

O trabalho demonstrou que técnicas de otimização exatas e heurísticas, quando bem calibradas e integradas, permitem ganhos concretos em logística ferroviária usando ferramentas open source. Apesar de a cobertura não ter atingido os 90 % almejados, o estudo apontou caminhos claros (rebalanceamento de frota, tuning de α, heurísticas híbridas) para reduzir o gap remanescente. O legado do TCC está no código replicável, na metodologia documentada e na agenda de pesquisa que pavimenta a adoção de modelos mais dinâmicos e robustos na prática operacional da MRS Logística S.A..

ANEXO I

```python
import pandas as pd
import math
from pyomo.environ import (
    ConcreteModel, Set, Param, Var, ConstraintList,
    NonNegativeIntegers, Objective, SolverFactory, maximize
)

# ---------- 1) Carrega planilhas ----------
file_path = "Arquivo de Parametros TCC.xlsx"
xls       = pd.ExcelFile(file_path)
df_fluxos = xls.parse("Parametros Fluxos").rename(columns=str.strip)
df_disp   = xls.parse("Disponibilidade").rename(columns=str.strip)\
                                        .set_index("Tipo Vagao")
df_vol    = xls.parse("Volume").rename(columns=str.strip)

# ---------- 2) Demanda (fluxo, semana) ----------
df_dem = df_vol.melt(id_vars=["Fluxo"],
    value_vars=[c for c in df_vol.columns if c.startswith("Sem")],
    var_name="Semana", value_name="Volume")
df_dem["Semana"] = df_dem["Semana"].str.extract(r"(\d+)").astype(int)
semanas = sorted(df_dem["Semana"].unique())
D = {(int(r.Fluxo), int(r.Semana)): float(r.Volume) for _, r in df_dem.iterrows()}

# ---------- 3) Fluxos e subtipos ----------
fluxos = sorted(df_fluxos["FLUXO"].astype(int))
subtipos = pd.unique(df_fluxos[["VAGAO TIPO 1","VAGAO TIPO 2"]].values.ravel())
subtipos = [s for s in subtipos if pd.notna(s)]

# ---------- 4) Capacidade util (TU) ----------
TU = {}
for _, row in df_fluxos.iterrows():
    f = int(row["FLUXO"])
```

```
    for col in ["TIPO 1","TIPO 2"]:
        nome = row.get(f"VAGAO {col}")
        tu   = row.get(f"TU POR VAGAO {col}")
        if pd.notna(nome) and pd.notna(tu):
            TU[(f, nome)] = float(tu)

# ---------- 5) Disponibilidade ----------
V = {(t, s): float(df_disp.loc[t, f"Sem {s}"])
     for t in subtipos for s in semanas}

# ---------- 6) Ciclo planejado (dias) ----------
ciclo_dias = df_fluxos.set_index("FLUXO")["CICLO PLANEJADO"]\
                      .fillna(7).astype(int).to_dict()

# ---------- 7) Viagens possiveis por semana ----------
trips_per_week = {f: max(1, math.floor(7.0 / ciclo_dias.get(f,
    7)))
                  for f in fluxos}

# ---------- 8) Upper bounds ----------
max_x = {}
for f in fluxos:
    for s in semanas:
        dem = D.get((f, s), 0.0)
        for t in subtipos:
            cap = TU.get((f, t), 0.0)
            mb  = math.ceil(dem / cap) if cap else 0
            max_x[(f, s, t)] = min(mb, int(V.get((t, s), 0)))

# ---------- 9) Parametro de penalidade ----------
alpha = 10.0

# ---------- 10) Modelo MIP ----------
model = ConcreteModel()
model.F = Set(initialize=fluxos)
model.S = Set(initialize=semanas)
model.T = Set(initialize=subtipos)

model.D  = Param(model.F, model.S, initialize=lambda m,f,s:
    D.get((f,s), 0.0))
```

```
model.V  = Param(model.T, model.S, initialize=lambda m,t,s:
   V.get((t,s), 0.0))
model.TU = Param(model.F, model.T, initialize=lambda m,f,t:
   TU.get((f,t), 0.0))
model.alpha = Param(initialize=alpha)

model.x = Var(model.F, model.S, model.T,
              domain=NonNegativeIntegers,
              bounds=lambda m,f,s,t: (0, max_x[(f,s,t)]))

# -- R1 Demanda
model.demanda = ConstraintList()
for f in fluxos:
    for s in semanas:
        model.demanda.add(
            sum(model.x[f,s,t] * model.TU[f,t] for t in
               subtipos) <= model.D[f,s])

# -- R2 Ciclo / viagens
model.ciclo = ConstraintList()
for f in fluxos:
    trips = trips_per_week[f]
    for s in semanas:
        lhs = sum(model.x[f,s,t] for t in subtipos)
        rhs = trips * sum(model.V[t,s] for t in subtipos)
        model.ciclo.add(lhs <= rhs)

# -- R3 Disponibilidade
model.dispo = ConstraintList()
for t in subtipos:
    for s in semanas:
        model.dispo.add(
            sum(model.x[f,s,t] for f in fluxos) <= model.V[t,s])

# ---------- 12) Funcao Objetivo ----------
model.obj = Objective(
    expr=(sum(model.x[f,s,t] * model.TU[f,t]
              for f in fluxos for s in semanas for t in subtipos)
          - model.alpha * sum(model.x[f,s,t]
```

```
                for f in fluxos for s in semanas for t in
                    subtipos)),
    sense=maximize)

# ---------- 13) Solver ----------
solver = SolverFactory("cbc")
solver.options["threads"] = 4
solver.solve(model, tee=False)

# ---------- 14) Exporta resultados ----------
resultado = []
atendida = demandada = 0.0
for f in fluxos:
    for s in semanas:
        demandada += model.D[f,s]
        for t in subtipos:
            q = model.x[f,s,t].value or 0
            if q > 0:
                tu = model.TU[f,t]
                atendida += q * tu
                resultado.append({
                    "Semana": s, "Fluxo": f, "Subtipo": t,
                    "Vagoes": int(q), "TU_vagao": tu,
                    "TU_atendida": q * tu, "TU_solicitada":
                        model.D[f,s]
                })

if resultado:
    df_res =
        pd.DataFrame(resultado).sort_values(["Semana","Fluxo",
    "Subtipo"])
    with pd.ExcelWriter("Resultado_Alocacao_Final.xlsx") as w:
        df_res.to_excel(w, "Alocacao Detalhada", index=False)
        pd.DataFrame({"Atendida":[atendida],
                      "Demandada":[demandada]})\
          .to_excel(w, "Resumo", index=False)
    print(f"OK  Atendida={atendida:.2f} t /
        Demandada={demandada:.2f} t")
else:
    print("WARNING  Nenhuma alocacao viavel encontrada.")
```

Listing 5.1 – Pipeline completo: carga de dados, modelo MIP e exportação

ANEXO II

```
!pip install -q deap pandas openpyxl numpy

import pandas as pd, numpy as np, random, math, time
from deap import base, creator, tools, algorithms

# ---------------------- 1) Leitura dos dados -------------------
FILE = "Arquivo de Parametros TCC.xlsx"
xls  = pd.ExcelFile(FILE)

df_fluxos = xls.parse("Parametros
   Fluxos").rename(columns=str.strip)
df_disp   =
   xls.parse("Disponibilidade").rename(columns=str.strip).
set_index("Tipo Vag o")
df_vol    = xls.parse("Volume").rename(columns=str.strip)

df_dem = (df_vol.melt(id_vars=["Fluxo"],
                      value_vars=[c for c in df_vol.columns if
                         c.startswith("Sem ")],
                      var_name="Semana", value_name="Volume")
                .assign(Semana=lambda d:
                   d["Semana"].str.extract(r"(\d+)").astype(int)))

FLUXOS   = sorted(df_fluxos["FLUXO"].astype(int))
SEMANAS  = sorted(df_dem["Semana"].unique())
SUBTIPOS = [s for s in pd.unique(df_fluxos[["VAG O TIPO
   1","VAG O TIPO 2"]].values.ravel()) if pd.notna(s)]

D  = {(int(r.Fluxo), int(r.Semana)): float(r.Volume) for _, r in
   df_dem.iterrows()}
TU = {(int(r["FLUXO"]), nome): float(r[f"TU POR VAG O {tipo}"])
      for _, r in df_fluxos.iterrows()
      for tipo, nome in zip(["TIPO 1","TIPO 2"], [r["VAG O TIPO
         1"], r["VAG O TIPO 2"]])
      if pd.notna(nome)}

TU_DEF = np.mean(list(TU.values()))
```

```
V = {(t,s): float(df_disp.loc[t, f"Sem {s}"]) for t in SUBTIPOS
    for s in SEMANAS}

ciclo = df_fluxos.set_index("FLUXO")["CICLO
    PLANEJADO"].fillna(7).astype(int).to_dict()
trips = {f: max(1, math.floor(7/ciclo.get(f,7))) for f in FLUXOS}
cap_fs = {(f,s): trips[f] * sum(V[(t,s)] for t in SUBTIPOS) for
    f in FLUXOS for s in SEMANAS}

# ---------------------- 2) GA      modelagem
    ------------------------
ALPHA   = 10.0    # penalidade por vag o
PEN_BIG = 1e3     # penalidade suave

genes = [(f,s,t) for f in FLUXOS for s in SEMANAS for t in
    SUBTIPOS if (f,t) in TU]
bounds = {(f,s,t): (0,
          min(math.ceil(D.get((f,s),0)/TU.get((f,t),TU_DEF)),
             int(V[(t,s)])))
          for (f,s,t) in genes}

# Classes DEAP
if "FitnessMin" not in creator.__dict__:
    creator.create("FitnessMin", base.Fitness, weights=(-1.0,))
if "Individual" not in creator.__dict__:
    creator.create("Individual", list,
       fitness=creator.FitnessMin)

toolbox = base.Toolbox()
toolbox.register("attr_int", lambda g:
    random.randint(*bounds[g]))
toolbox.register("individual", tools.initIterate,
    creator.Individual,
                 lambda: [toolbox.attr_int(g) for g in genes])
toolbox.register("population", tools.initRepeat, list,
    toolbox.individual)

# --------------------- Reparo coerente -------------------------
def repair(q_dict):
    # R3: disponibilidade por subtipo/semana
```

```
62      for t in SUBTIPOS:
63          for s in SEMANAS:
64              uso = sum(q_dict.get((f,s,t),0) for f in FLUXOS)
65              if uso > V[(t,s)]:
66                  excesso = uso - V[(t,s)]
67                  cand_flux = [ff for ff in FLUXOS if (ff,s,t) in
                        q_dict and q_dict[(ff,s,t)]>0]
68                  cand_flux.sort(key=lambda ff:
                        TU.get((ff,t),TU_DEF))  # tira primeiro os
                        menos eficientes
69                  for ff in cand_flux:
70                      if excesso <= 0: break
71                      take = min(excesso, q_dict[(ff,s,t)])
72                      q_dict[(ff,s,t)] -= take
73                      excesso -= take
74
75      # R2: limite de "viagens equivalentes" por fluxo/semana
76      for f in FLUXOS:
77          for s in SEMANAS:
78              soma_vag = sum(q_dict.get((f,s,t),0) for t in
                    SUBTIPOS if (f,s,t) in q_dict)
79              if soma_vag > cap_fs[(f,s)]:
80                  excesso = soma_vag - cap_fs[(f,s)]
81                  cand_t = [tt for tt in SUBTIPOS if (f,s,tt) in
                        q_dict and q_dict[(f,s,tt)]>0]
82                  cand_t.sort(key=lambda tt:
                        TU.get((f,tt),TU_DEF))  # tira primeiro os
                        menos TU
83                  for tt in cand_t:
84                      if excesso <= 0: break
85                      take = min(excesso, q_dict[(f,s,tt)])
86                      q_dict[(f,s,tt)] -= take
87                      excesso -= take
88      return q_dict
89
90  # --------------------- Fitness -------------------------------
91  def evaluate(ind):
92      q = {g: ind[i] for i,g in enumerate(genes)}
93      q = repair(q)  # garante factibilidade (ou quase)
94
```

```
95      corte = excesso = 0.0
96      pen   = 0.0
97
98      # R1: corte / excesso
99      for f in FLUXOS:
100         for s in SEMANAS:
101             atend = sum(q.get((f,s,t),0)*TU.get((f,t),TU_DEF)
                    for t in SUBTIPOS if (f,t) in TU)
102             dem   = D.get((f,s),0.0)
103             if atend < dem: corte   += dem - atend
104             else:           excesso += atend - dem
105
106     # p s-reparo, viola  es devem ser raras, mas penaliza se
        existirem
107     for t in SUBTIPOS:
108         for s in SEMANAS:
109             uso = sum(q.get((f,s,t),0) for f in FLUXOS)
110             if uso > V[(t,s)]:
111                 pen += (uso - V[(t,s)]) * PEN_BIG
112
113     for f in FLUXOS:
114         for s in SEMANAS:
115             soma_vag = sum(q.get((f,s,t),0) for t in SUBTIPOS if
                    (f,s,t) in q)
116             if soma_vag > cap_fs[(f,s)]:
117                 pen += (soma_vag - cap_fs[(f,s)]) * PEN_BIG
118
119     vag_total = sum(q.values())
120     return (corte + excesso + ALPHA*vag_total + pen,)
121
122 toolbox.register("evaluate", evaluate)
123 toolbox.register("mate", tools.cxTwoPoint)
124
125 def mut_inplace(ind, indpb):
126     for i,g in enumerate(genes):
127         if random.random() < indpb:
128             ind[i] = random.randint(*bounds[g])
129     return (ind,)
130
131 toolbox.register("mutate", mut_inplace, indpb=0.03)
```

```
132 toolbox.register("select", tools.selTournament, tournsize=3)
133
134 # ---------------------- 3) Execucao GA --------------------------
135 POP  = 300
136 NGEN = 600
137 CXPB, MUTPB = 0.85, 0.10
138 SEED = 123
139
140 random.seed(SEED)
141 pop = toolbox.population(POP)
142 hof = tools.HallOfFame(1)
143
144 print("===> Evoluindo GA restrito (com reparo)   ")
145 t0 = time.time()
146 BEST_FIT = 1e30
147 no_improv = 0
148
149 for gen in range(1, NGEN+1):
150     pop = algorithms.varAnd(pop, toolbox, cxpb=CXPB, mutpb=MUTPB)
151     invalid = [ind for ind in pop if not ind.fitness.valid]
152     fits = map(toolbox.evaluate, invalid)
153     for ind, fit in zip(invalid, fits):
154         ind.fitness.values = fit
155
156     pop = toolbox.select(pop, k=len(pop))
157     hof.update(pop)
158
159     cur_best = min(ind.fitness.values[0] for ind in pop)
160     if cur_best + 1e-6 < BEST_FIT:
161         BEST_FIT = cur_best
162         no_improv = 0
163     else:
164         no_improv += 1
165
166     if gen % 50 == 0:
167         avg_fit = np.mean([i.fitness.values[0] for i in pop])
168         print(f"Gera  o {gen:4d} | best={cur_best:,.2f} |
              avg={avg_fit:,.2f}")
169
170     if no_improv >= 200:
```

```
171         print(f"-> Parando cedo em {gen} (sem melhoria).")
172         break
173
174 print(f"Tempo GA: {time.time()-t0:0.1f}s")
175
176 # ---------------------- 4) Resultado ----------------------------
177 best = hof[0]
178 qbest = {g: best[i] for i,g in enumerate(genes)}
179 qbest = repair(qbest)
180
181 atendida   = sum(qbest[g] * TU[g[0], g[2]] for g in qbest)
182 demandada  = sum(D.values())
183 pct        = 100*atendida/demandada
184
185 print(f"Atendida = {atendida:,.0f} t / Demandada =
    {demandada:,.0f} t ({pct:5.2f} %)")
186
187 df_ga = pd.DataFrame(
188     [{"Semana":g[1], "Fluxo":g[0], "Subtipo":g[2],
189       "Vag es":qbest[g],
190       "TU atendida":qbest[g]*TU[(g[0],g[2])]}
191      for g in genes if qbest[g] > 0]
192 ).sort_values(["Semana","Fluxo","Subtipo"])
193
194 df_ga.to_excel("Resultado_AG_MIPrestrito.xlsx", index=False)
195 print("Arquivo 'Resultado_AG_MIPrestrito.xlsx' salvo.")
```

Listing 5.2 – Pipeline completo: carga de dados, modelo DEAP e exportação

REFERÊNCIAS

Agência de Notícias da Indústria. **Brasil e a Dependência das Rodovias: Desafios e Caminhos para Investimentos**. 2024. Acesso em: 26 mai. 2025. Disponível em: <https://noticias.portaldaindustria.com.br/noticias/infraestrutura/brasil-e-a-dependencia-das-rodovias-desafios-e-caminhos-para-investimentos/>.

Agência Nacional de Transportes Terrestres (ANTT). **Relatório de Desempenho do Setor Ferroviário**. 2024. Acesso em: mai. 2025. Disponível em: <https://www.antt.gov-.br>.

ALMEIDA, Beatriz C. **Hybrid BRKGA for the Berth Allocation Problem**. Dissertação (Mestrado) — Universidade de Lisboa, Lisboa, 2021.

CORDEAU, Jean-François; LAPORTE, Gilbert; MERCIER, André. The railcar assignment problem: Overview and industrial applications. **Transportation Research Part B: Methodological**, v. 34, n. 3, p. 171–189, 2000.

DESAULNIERS, Guy; LANGEVIN, Alain. A genetic algorithm for capacity allocation in railway networks. **Computers & Operations Research**, v. 47, p. 1–12, 2014.

GUTH, Gabriel Durães. **Análise dos Métodos de Resolução de Problemas de Programação Inteira: Estudo de Caso na ANAC**. 2021.

Instituto Aço Brasil. **Anuário Estatístico do Aço 2025**. 2025. Acesso em: 6 jul. 2025. Disponível em: <https://acobrasil.org.br/anuario2025>.

LIU, Xuan; WANG, Zhihao. Slotting optimization in rail yards using continuous genetic algorithms. **Journal of Rail Transport Planning & Management**, v. 21, p. 100–112, 2023.

MRS Logística. **Relatório de Desempenho Operacional 2023**. 2023. Acesso em: 6 jul. 2025. Disponível em: <https://www.mrs.com.br/relatorios/operacional2023.pdf>.

MRS Logística S.A. **Ferrovia e Frota**. 2025. `https://www.mrs.com.br/empresa/ferrovia-frota/`. Acesso em 21 jul. 2025.

NAZRI, S. N. A.; OUTROS. Optimization berth allocation in container terminals: A pyomo and google colab approach. **Ocean & Coastal Management**, 2024.

OLIVEIRA, Marina C. **Modelagem e Solução de um Problema de Alocação de Contêineres Utilizando Pyomo e CBC**. Dissertação (Mestrado) — Universidade Federal do Espírito Santo, Vitória, 2020.

SILVA, Marlon R.; SANTOS, João F. An adaptive genetic algorithm for vehicle routing with time windows. In: **Proceedings of the 15th Metaheuristics Conference**. [S.l.: s.n.], 2019. p. 122–129.

SOUZA, Leonardo B. de. **Aplicação de Programação Inteira Mista na Otimização do Carregamento de Caminhões em uma Operadora Logística**. Belo Horizonte: [s.n.], 2018.

TAGARO, Maria; FERNANDEZ, Luis; SILVA, Eduardo. Heurísticas evolutivas em cadeias de suprimento dinâmicas: Revisão e perspectivas. **Journal of Logistics and Supply Chain Management**, v. 12, n. 1, p. 45–67, 2024.

TOSTES, Pedro Duarte Gomes. **Metodologia para Dimensionamento de Frota da Grade de Carga Geral**. Dissertação (Mestrado) — PUC–Rio, 2022.

ZHANG, Ying; LI, Wen; ZHOU, Hao. Forecasting seasonal freight demand using generalized regression neural networks and genetic algorithms. **Transportation Research Part C: Emerging Technologies**, v. 68, p. 113–128, 2016.