Otimização Contínua e Combinatória

Prof. Dr. Bruno Costa e Silva Nogueira Redhttp://professor.ufal.br/bruno/

Prof. Dr. Rian Gabriel S. Pinheiro Redhttp://professor.ufal.br/rian/

2025.1

Wave Order Picking PLIwC²

Uma aplicação PLI com CPLEX & CUDA para o Desafio SBPO 2025

Fábio Linhares Red`fl@email.com` **Hans Ponfick** Red`hapl@email.com`

Resumo

Este artigo apresenta o desenvolvimento e a análise de uma abordagem de otimização exata, fundamentada em Programação Linear Inteira (PLI) e significativamente acelerada por GPU com NVIDIA CUDA, para a resolução do Problema de Wave Order Picking (WOP) proposto no Desafio SBPO 2025 do Mercado Livre. O WOP, um problema de natureza NP-Difícil, consiste na seleção otimizada de um subconjunto de pedidos (denominado wave) e um subconjunto de corredores a serem visitados em um centro de distribuição, com o objetivo de maximizar a eficiência da coleta. Detalhamos a formulação matemática do problema, incluindo a complexa tarefa de linearizar uma função objetivo fracionária, para a qual exploramos três métodos distintos: Variável Inversa, Charnes-Cooper e o algoritmo iterativo de Dinkelbach. Discutimos a implementação de um modelo flexível que acomoda tanto restrições rígidas quanto suaves (com penalidades), a aplicação de técnicas avançadas de pré-processamento de dados e do modelo matemático para redução do espaço de busca, e a crucial integração com o solver IBM CPLEX e a tecnologia CUDA. Esta combinação permitiu obter resultados altamente competitivos, igualando a solução ótima em 14 a 16 das 20 instâncias do desafio, dentro de limites de tempo exíguos (aproximadamente 10 minutos por instância). Uma análise detalhada e passo a passo de uma instância de exemplo (t/0001) é fornecida para ilustrar a aplicação prática e os meandros do modelo. Os resultados experimentais e a discussão subsequente demonstram a viabilidade e a eficácia da abordagem exata quando sinergicamente combinada com modelagem matemática sofisticada e aceleração computacional de ponta, oferecendo uma alternativa poderosa às tradicionais abordagens heurísticas para este complexo problema logístico.

Palavras-chave: Wave Order Picking, Otimização Exata, Programação Linear Inteira (PLI), Aceleração por GPU (CUDA), Linearização de Função Fracionária, CPLEX, Pré-processamento, Logística de E-commerce.

1 Nota dos autores

Este artigo foi elaborado como parte dos requisitos avaliativos da disciplina de Otimização Contínua e Combinatória, componente curricular do Mestrado em Informática da Universidade Federal de Alagoas (UFAL). A proposta consistiu em pesquisar, analisar e implementar uma solução para um problema real e complexo de otimização combinatória, apresentando os resultados em formato de artigo científico. Para isso, escolhemos trabalhar com o Wave Order Picking, um problema logístico proposto pelo Mercado Livre no Desafio SBPO 2025.

Como a implementação computacional era obrigatória, optamos inicialmente por desenvolver uma abordagem heurística, considerando a alta complexidade do problema — classificado como NP-Difícil. No entanto, decidimos avançar para uma abordagem exata, utilizando Programação Linear Inteira (PLI), com aceleração por GPU via NVIDIA CUDA. Essa escolha mostrou-se não apenas viável, como também extremamente eficaz, permitindo alcançar soluções ótimas em tempos computacionalmente aceitáveis para as instâncias do desafio.

Adotamos, ao longo do texto, um estilo conscientemente didático e acessível, sem abrir mão do rigor técnico exigido pela natureza do tema. Nossa intenção foi tornar o conteúdo compreensível não apenas para especialistas em otimização e ciência da computação, mas também para leitores de áreas correlatas ou interessados no assunto, mesmo sem formação técnica aprofundada. Esperamos, com isso, contribuir para uma divulgação mais ampla do conhecimento, equilibrando clareza, precisão conceitual e profundidade analítica.

2 Introdução

A eficiência nas operações logísticas dos centros de distribuição tornou-se um diferencial competitivo crucial no setor varejista, especialmente no comércio eletrônico (GOSHIP, 2024). Nesse contexto, a logística transcende o mero transporte de mercadorias, abarcando uma gama de processos internos essenciais à operação eficaz desses centros.

Dentre esses processos, destaca-se a separação de pedidos, conhecida como *order picking*. Essa etapa envolve a localização e coleta dos itens solicitados pelos clientes no estoque e pode representar entre 50% e 65% dos custos operacionais totais em centros de distribuição (BOZ, E.; ARAS, N., 2022; ÁGUIA SISTEMAS, 2024). Além do impacto financeiro, a atividade consome cerca de 55% do tempo total das operações logísticas (BOZ, E.; ARAS, N., 2022), afetando diretamente indicadores críticos, como o tempo de ciclo do pedido — intervalo entre o recebimento da ordem e sua expedição — e a acurácia das entregas (VTEX, 2020).

Para mitigar esses custos e tempos, diversas abordagens vêm sendo propostas com o objetivo de otimizar o processo de separação de pedidos. Dentre elas, destaca-se o *Order Batching Problem* (OBP), que consiste na formação de grupos de pedidos a serem coletados conjuntamente, visando à minimização do tempo ou da distância total percorrida pelos coletores, bem como ao balanceamento da carga de trabalho. O OBP é, portanto, um subproblema fundamental dentro do processo mais amplo de *order picking* (BOZ, E.; ARAS, N., 2022).

Uma variação relevante dessa abordagem é o *Wave Order Picking* (WOP), ou separação de pedidos por ondas, que agrupa pedidos com características similares para processamento simultâneo, buscando maximizar a eficiência da coleta. Nesse caso, a produtividade pode ser expressa pela razão entre o número total de itens coletados e o número de corredores efetivamente visitados no armazém (NETSUITE, 2025; INVIA ROBOTICS, 2024; SOUZA, 2025). Estimativas indicam que essa abordagem pode reduzir em até 40% o tempo total de deslocamento dos coletores, em comparação com métodos tradicionais de coleta individual (BOZ, E.; ARAS, N., 2022; NETSUITE, 2025).

A eficácia do WOP depende da qualidade dos agrupamentos formados. Critérios como a similaridade entre os itens, prazos de entrega, rotas internas e destinos comuns devem ser considerados na formação das ondas, com o objetivo de evitar deslocamentos redundantes e otimizar a utilização de recursos logísticos — humanos (como a força de trabalho) e físicos (como carrinhos, esteiras e leitores de código de barras) (NETSUITE, 2025; INVIA ROBOTICS, 2024).

Embora o conceito de WOP seja intuitivo, seu tratamento computacional revela uma estrutura altamente complexa. O problema envolve decisões de agrupamento (*order batching*) e, implicitamente, de roteamento dos coletores (*Picker Routing Problem — PRP*), caracterizando-o como um problema de natureza combinatória *NP-difícil* (BOZ, E.; ARAS, N., 2022), o que significa que sua complexidade cresce exponencialmente com o aumento da instância. Na prática, essa característica costuma tornar a aplicação de métodos exatos inviáveis para grandes volumes de dados dentro de prazos computacionalmente aceitáveis. Talvez por isso as pesquisas priorizem o uso de abordagens heurísticas e meta-heurísticas, como apurado em levantamento recente: 51% dos estudos sobre OBP utilizam meta-heurísticas, enquanto apenas 17% recorrem a métodos exatos, especialmente quando o problema é dinâmico ou está integrado a outras decisões operacionais (BOZ, E.; ARAS, N., 2022).

O problema proposto pelo Mercado Livre no Desafio SBPO 2025 (SOUZA, 2025; PESQUISA OPERACIONAL, 2025; MERCADO LIVRE, 2025) insere-se precisamente nesse contexto, desafiando os participantes a maximizar a razão entre o número total de unidades dos pedidos selecionados e o número de corredores efetivamente percorridos. Contrariando a predominância de abordagens exclusivamente heurísticas e meta-heurísticas na literatura, este trabalho propõe enfrentar o problema com uma abordagem exata baseada em Programação Linear Inteira (PLI). Demonstra-se que, mesmo diante da complexidade combinatória inerente ao *Wave Order Picking* (WOP), é possível obter soluções ótimas, com prova formal de otimalidade, dentro dos prazos computacionalmente viáveis estabelecidos pelo Desafio SBPO 2025 (SOUZA, 2025; PESQUISA OPERACIONAL, 2025). Mas para tornar essa abordagem viável em tempo hábil, optamos por uma conjunção estratégica de técnicas de modelagem, pré-processamento e aceleração computacional, organizadas em alguns pilares principais:

1. **Linearização da função objetivo fracionária:** a função que expressa a produtividade como a razão entre itens coletados e corredores percorridos é intrinsecamente não linear. Para torná-la tratável por solvers de PLI, foram avaliadas e comparadas três estratégias matemáticas: o método da variável inversa, a transformação de Charnes-Cooper e o algoritmo iterativo de Dinkelbach. A primeira e a última abordagens, em especial, viabilizam a modelagem que idealizamos.

2. **Pré-processamento estruturado:** para organizar os dados de entrada, eliminar redundâncias e reduzir significativamente o espaço de busca desenvolvemos um pré-processamento estruturado que inclui a identificação de pedidos e corredores relevantes, a eliminação de pedidos inviáveis e a organização dos dados em estruturas otimizadas para acesso rápido durante a resolução que minimizam o tempo de computação.

3. **Integração com o solver CPLEX:** a implementação do modelo matemático foi realizada utilizando o solver IBM ILOG CPLEX, que oferece suporte robusto para PLI e é amplamente reconhecido por sua eficiência em problemas combinatórios complexos e essa integração permitiu explorar ao máximo as capacidades de otimização exata, garantindo soluções ótimas e verificáveis.

4. **Aceleração computacional com CUDA:** para lidar com a alta demanda computacional utilizamos o processamento massivamente paralelo da arquitetura NVIDIA CUDA e, com isso, conseguimos acelerar significativamente as etapas de pré-processamento e avaliação de restrições, reduzindo o tempo de execução e permitindo a resolução dentro dos limites temporais do desafio (10 minutos por instância). A aceleração foi implementada em três frentes principais:

- **Pré-processamento paralelo:** o cálculo das interseções entre pedidos e corredores, uma operação tipicamente de complexidade $O(n^2)$, foi paralelizado, reduzindo-a para $O(n/k)$, onde k representa o número de núcleos CUDA disponíveis, o que otimiza a preparação dos dados para o *solver*.
- **Avaliação paralela de restrições:** milhares de combinações de pedidos são verificadas simultaneamente quanto à viabilidade, acelerando a exploração de soluções candidatas.
- **Geração paralela da matriz de coeficientes:** especialmente eficiente para instâncias esparsas e de grande porte, esta etapa distribui a construção dos coeficientes do modelo entre os núcleos CUDA, reduzindo drasticamente o tempo de compilação do modelo matemático antes de sua submissão ao solver.

Testes empíricos indicaram uma aceleração média de 28x em relação à versão sequencial em CPU para instâncias com mais de 200 pedidos, sendo esse ganho crucial para o cumprimento dos rigorosos limites temporais do desafio.

5. **Modelagem flexível com restrições rígidas e suaves:** o modelo admite tanto restrições estritamente obrigatórias quanto restrições relaxáveis, que podem ser violadas mediante aplicação de penalidades na função objetivo. Essa flexibilidade permite capturar nuances operacionais e adaptar a formulação às diferentes exigências das instâncias reais, oferecendo maior robustez e aplicabilidade.

Em conjunto, essas técnicas viabilizam uma solução exata competitiva, sob as restrições temporais do desafio. Este artigo portanto, se propõe a detalhar a formulação matemática do WOP, as técnicas de linearização e pré-processamento empregadas, a arquitetura da solução com ênfase na integração com CPLEX e CUDA, e apresentar uma análise de desempenho criteriosa, baseada nos resultados obtidos em um conjunto representativo de instâncias do Desafio SBPO 2025. O trabalho visa demonstrar que, munidos das ferramentas conceituais e tecnológicas adequadas, os métodos exatos não apenas são viáveis, mas podem se mostrar altamente competitivos e oferecer soluções de qualidade para problemas combinatórios complexos no domínio da logística de armazéns, bem como contribuir com um arcabouço metodológico que pode ser estendido a outros problemas de natureza combinatória computacionalmente exigentes.

3 Justificativa e Relevância

A otimização do processo de *picking* é crucial, pois pode representar mais de 50% dos custos operacionais totais de um armazém(BOZ, E.; ARAS, N., 2022). O *Wave Order Picking* (WOP), ao organizar a coleta em ondas discretas, busca melhorar a eficiência, reduzir gargalos e garantir o envio pontual dos pedidos(NETSUITE, 2025). No ambiente competitivo do *e-commerce*, como o operado pelo Mercado Livre, a capacidade de processar um alto volume de pedidos de forma rápida e precisa é um diferencial estratégico(INVIA ROBOTICS, 2024). Desenvolver um modelo exato para o WOP, adaptado às especificidades do Desafio SBPO 2025, permite uma compreensão profunda dos *trade-offs* envolvidos e pode levar a economias significativas e melhorias no nível de serviço.

A natureza combinatória do WOP, que integra aspectos de *order batching* (agrupamento de pedidos) e *picker routing* (roteirização de coletores), torna sua otimização um desafio considerável (POCINHO, G. F. C., 2013). Enquanto métodos heurísticos e meta-heurísticos são frequentemente empregados para obter boas soluções em tempo computacional razoável para grandes instâncias (DCC/UFMG, s.d.), os métodos exatos são fundamentais para:

- Fornecer soluções ótimas para instâncias de tamanho moderado;

- Avaliar a qualidade das soluções obtidas por heurísticas;
- Obter *insights* estruturais das soluções ótimas que podem guiar o desenvolvimento de heurísticas mais eficazes.

Este estudo se justifica (não pela necessidade de aprovação na disciplina, claro que não!) pela necessidade de explorar o potencial máximo de otimização para o WOP dentro das condições do desafio, utilizando uma abordagem rigorosa de modelagem matemática.

4 Objetivos do Estudo

O objetivo principal deste trabalho é desenvolver, implementar e avaliar um modelo de Programação Linear Inteira (*Integer Linear Programming* — ILP) para a resolução exata do Problema de Wave Order Picking, conforme definido no Desafio SBPO 2025 do Mercado Livre.

Os objetivos específicos incluem:

- Realizar uma revisão do estado da arte sobre o WOP, o *Order Batching Problem*, modelos matemáticos e algoritmos de solução;
- Formular um modelo ILP detalhado que capture as características específicas do problema do desafio, incluindo a formação de ondas, capacidade dos coletores, agrupamento de pedidos e sequenciamento das operações;
- Implementar o modelo ILP utilizando um *CPLEX* com *CUDA* para acelerar o processo de resolução, aproveitando a capacidade de processamento paralelo das GPUs;
- Analisar o desempenho da solução exata em termos de qualidade da solução e esforço computacional, utilizando as instâncias e o validador fornecidas pelo desafio;
- Identificar as limitações da abordagem exata e propor direções para trabalhos futuros.

5 Estrutura do Artigo

Este artigo está organizado da seguinte forma:

6 Revisão da Literatura

A otimização do *Wave Order Picking* (WOP), e de seu subproblema central, o *Order Batching Problem* (OBP), tem sido objeto de intensa investigação nas últimas décadas. A dificuldade inerente ao OBP, classificado como um problema NP-difícil (GADEMANN; VELDE, 2005; BOZ, Emrah; ARAS, Necati, 2022), torna a obtenção de soluções ótimas em tempo razoável impraticável para instâncias de escala realista. Por essa razão, a maior parte da literatura recorre a abordagens heurísticas e meta-heurísticas, com o objetivo de obter soluções de boa qualidade com eficiência computacional. Na contramão, vamos tentar explorar as formulações fracionárias e suas potenciais vantagens no contexto do WOP e, com sorte, abrir caminho para investigações mais inovadoras no campo.

6.1 Etapas Operacionais e Variações do WOP

O processo de *wave picking* em ambientes logísticos é, tipicamente, dividido em três fases operacionais bem estabelecidas (MECALUX, 2020; E-SHIP WMS, 2024; PLATINUM LOG, 2023):

1. **Pré-onda**
   Envolve o planejamento e o agendamento das ondas, incluindo o agrupamento de pedidos segundo critérios logísticos, tais como proximidade geográfica no layout, prioridades de entrega ou restrições de tempo.

2. **Execução da onda**
   Corresponde à etapa de coleta física dos itens nos armazéns, geralmente suportada por sistemas de gerenciamento de armazém (WMS) e tecnologias de identificação automática. Os coletores percorrem corredores e prateleiras, coletando múltiplos pedidos por onda.

3. **Pós-onda**
   Abrange a consolidação dos itens coletados, a separação dos pedidos individuais e a preparação para expedição, frequentemente integrada a sistemas de transporte e distribuição.

Destacam-se, em especial, duas variantes operacionais com impacto direto na modelagem matemática do problema (MECALUX, 2020; CALAFATE, 2020):

- **Fixed wave picking:** todos os pedidos de uma onda são coletados e consolidados simultaneamente antes da expedição;

- **Dynamic wave picking:** pedidos individuais podem ser liberados para expedição tão logo estejam prontos, otimizando o fluxo de saída e reduzindo o tempo de ciclo total.

Apesar da clareza das etapas operacionais, nota-se que grande parte da literatura técnica ignora nuances práticas da operação, como a influência de políticas de reabastecimento dinâmico ou interferências causadas por múltiplos coletores, o que limita a aderência dos modelos a cenários reais (MECALUX, 2020; E-SHIP WMS, 2024; PLATINUM LOG, 2023).

6.2 Heurísticas e Meta-heurísticas

Entre os métodos heurísticos, destacam-se algoritmos construtivos simples, como o *First-Come-First-Served* (FCFS), algoritmos baseados em sementes (*seed algorithms*) e heurísticas de roteirização, como a *S-shape* (HENN; KOCH; DOERNER et al., 2010). Essas abordagens são valorizadas por sua implementação direta e por apresentarem desempenho competitivo em cenários com complexidade moderada.

As meta-heurísticas têm se mostrado particularmente eficazes em ambientes mais complexos e dinâmicos. Estratégias como *Iterated Local Search* (ILS), *Ant Colony Optimization* (ACO), *Tabu Search* (TS) e *Genetic Algorithms* (GA) demonstram notável versatilidade e escalabilidade (MEDEIROS, 2023; TAVARES, 2020; DESCONHECIDO, 2018; ÖNCAN, Temel, 2013).

Além disso, abordagens híbridas que integram algoritmos de agrupamento e roteirização — como em (HENN; KOCH; WÄSCHER, 2012) — têm mostrado reduções significativas no *makespan* e ganhos operacionais relevantes. No entanto, poucos achados investigam o desempenho dessas heurísticas à luz de métricas alternativas de qualidade, como a razão entre unidades coletadas e deslocamento — precisamente o foco inovador deste trabalho.

Por fim, ressalta-se que, embora os modelos heurísticos geralmente sigam etapas bem definidas, parte da literatura tende a simplificar ou negligenciar aspectos práticos complexos, como políticas de reabastecimento dinâmico, restrições operacionais e interferências entre coletores. Não que isso tenha implicações diretas ou indiretas com nosso estudo, mas ainda assim os autores acreditam que tais omissões podem comprometer a aplicabilidade dos modelos a cenários reais (MECALUX, 2020; E-SHIP WMS, 2024; PLATINUM LOG, 2023).

6.3 Formulações Exatas e Benchmarks

Apesar do predomínio das heurísticas na literatura sobre o Order Batching Problem (OBP) e o Wave Order Picking (WOP), as formulações exatas desempenham um papel central tanto na construção de benchmarks quanto na compreensão estrutural das soluções ótimas. Modelos baseados em Programação Linear Inteira (ILP) e Programação Linear Inteira Mista (MILP) são amplamente utilizados nesse contexto.

As formulações clássicas concentram-se majoritariamente na minimização do *makespan* ou do tempo total de coleta. No entanto, essa ênfase acaba por negligenciar outras métricas relevantes para a eficiência logística contemporânea, como razões fracionárias, indicadores de sustentabilidade ou medidas de balanceamento. Essa limitação evidencia uma lacuna ainda pouco explorada na modelagem exata do problema.

Pansart et al. (PANSART; DAUZÈRE-PÉRÈS; GOURGAND, 2018) propuseram formulações MILP esparsas, reforçadas por técnicas de pré-processamento e desigualdades válidas, que se tornaram referência como benchmark para comparação de heurísticas. De forma complementar, Öncan (ÖNCAN, T., 2015) desenvolveu modelos MILP que incorporam diferentes políticas de roteirização, ampliando a aplicabilidade dos métodos exatos a diferentes configurações operacionais.

Contribuições mais especializadas também se destacam. O trabalho de Çağırıcı (ÇAĞRC, 2014) apresentou formulações MILP adaptadas a políticas específicas de movimentação no armazém, como *S-shape* e retorno, voltadas a sistemas de armazenagem de baixo nível. Já Shiau e Huang (SHIAU; HUANG, 2020) estenderam a modelagem exata ao contexto do *wave planning*, integrando o OBP com restrições de janelas de tempo e estratégias *just-in-time*, características comuns em ambientes de alta rotatividade e exigência logística.

Além dessas abordagens tradicionais, há um movimento crescente em direção a modelos multiobjetivo na otimização logística. Tais modelos conciliam critérios conflitantes — como o balanceamento de carga entre ondas e a minimização do percurso total — permitindo análises mais nuançadas dos trade-offs operacionais e promovendo maior aderência dos modelos às necessidades reais de planejamento (MARLER; ARORA, 2004; MIETTINEN, 1999).

Ainda assim, são raros os estudos que consideram métricas fracionárias como função objetivo principal — por exemplo, razões entre unidades coletadas e distância percorrida. Essa lacuna motiva a proposta deste trabalho, que explora tais razões como critério central de qualidade logística e analisa os desafios teóricos e computacionais envolvidos em sua modelagem e resolução (BERTSIMAS; SIM, 2011; BEN-TAL; EL GHAOUI; NEMIROVSKI, 2009).

6.4 Formulações Fracionárias: Teoria e Aplicações

O problema proposto no Desafio SBPO 2025 introduz uma função objetivo fracionária voltada à maximização da razão entre o número de unidades coletadas e o número de corredores percorridos. Essa modelagem insere o problema na classe da Programação Fracionária Inteira Mista (MILP fracionária), menos explorada no contexto do Wave Order Picking (WOP).

Há, entretanto, uma base teórica consolidada para esse tipo de formulação. Duas abordagens clássicas destacam-se: (i) a transformação de Charnes-Cooper (CHARNES; COOPER, 1962), que permite linearizar problemas com razão de funções lineares, sob certas condições; e (ii) o algoritmo de Dinkelbach (DINKELBACH, 1967), um método iterativo que resolve uma sequência de problemas paramétricos equivalentes, com boa eficiência para instâncias de grande porte.

Em casos onde o denominador é uma variável inteira — como o número de corredores visitados —, estratégias específicas, como relaxações convexas ou propriedades combinatórias de quocientes inteiros positivos, também podem ser empregadas. Um exemplo disso é o trabalho de Zhu e Wang (ZHU, 2016), que aplicam essas ideias ao contexto de problemas de corte em grafos.

Apesar da robustez teórica, aplicações dessas técnicas ao WOP ainda são praticamente inexistentes, o que revela uma lacuna importante a ser explorada. Novas investigações podem examinar como diferentes políticas de roteamento impactam a função objetivo fracionária, resultando em métricas mais alinhadas à produtividade operacional e contribuindo para o desenvolvimento de benchmarks mais representativos da realidade dos armazéns.

6.5 Técnicas de Linearização em MILP

A formulação de funções objetivo fracionárias ou restrições não lineares em modelos MILP exige técnicas de linearização eficazes, tanto para garantir a tratabilidade do problema quanto para manter o desempenho computacional aceitável. No contexto do Wave Order Picking (WOP), cuja proposta neste trabalho envolve a maximização da razão entre unidades coletadas e corredores percorridos, essas técnicas tornam-se especialmente relevantes.

As principais estratégias de linearização incluem:

- **Desigualdades de McCormick** (SHERALI; FRATICELLI, 2001): utilizadas para linearizar produtos entre variáveis binárias e contínuas. São amplamente empregadas por sua simplicidade e por não exigirem parâmetros arbitrários, embora nem sempre gerem limites duais apertados.
- **Método do Big-M** (WOLSEY, Laurence A., 1999): cria restrições condicionais ativadas por variáveis binárias. Apesar de sua versatilidade, requer o uso cuidadoso de constantes grandes (o "M"), cuja escolha afeta fortemente a estabilidade numérica e a eficiência da resolução.
- **Codificação binária e modelagem recíproca**: em problemas com denominadores inteiros (como o número de corredores), é possível representar o inverso de uma variável por meio de codificações binárias e variáveis auxiliares, ativadas seletivamente. Essa abordagem generaliza técnicas como o *Inverse Variable Linearizer* usado neste trabalho, permitindo representar expressões do tipo $1/\sum_i x_i$ sem recorrer a não linearidades explícitas.
- **Programação disjuntiva e reforço polinomial** (TAWARMALANI; SAHINIDIS, 2002; BALAS, 1998): essas técnicas buscam representar o espaço viável como união de politopos, permitindo a obtenção de relaxações mais fortes. No entanto, frequentemente resultam em modelos mais complexos, com maior número de variáveis e restrições.

Apesar da diversidade de métodos, parece haver menos estudos comparativos que avaliem, de forma sistemática, o impacto dessas técnicas no desempenho computacional e na qualidade das soluções do que se esperava, sobretudo em contextos com objetivo fracionário, como o WOP. Em particular, o trade-off entre a força da relaxação contínua, a estabilidade numérica e o custo de modelagem permanece pouco explorado na literatura. Não por isso, mas coincidentemente, a proposta deste trabalho é investigar essas questões, avaliando a eficácia de diferentes técnicas de linearização no contexto do WOP e propondo uma abordagem que combina o método do Big-M com a linearização recíproca, visando maximizar a eficiência computacional e a qualidade das soluções.

6.6 Relaxação Lagrangeana e Métodos Aumentados

Os métodos baseados em relaxação Lagrangeana têm se mostrado particularmente eficazes em problemas de otimização combinatória com estrutura complexa, como o Wave Order Picking (WOP). A ideia central consiste em incorporar restrições complicadas à função objetivo por meio de multiplicadores de Lagrange, decompondo o problema original em subproblemas mais simples e tratáveis. Essa abordagem tem sido empregada com sucesso em variantes do Order Batching and Picking (OBP), especialmente quando há acoplamento entre variáveis causado por restrições de capacidade ou janelas de tempo (FISHER, 2004; GEOFFRION, 1974; GUIGNARD, 2003).

Uma das principais vantagens da relaxação Lagrangeana reside na capacidade de gerar limites duais de alta qualidade. Tais limites são essenciais para avaliar a eficácia de heurísticas, especialmente em instâncias de grande porte onde a solução ótima não é conhecida. O Método do Lagrangeano Aumentado (Augmented Lagrangian Method – ALM), por sua vez, representa uma extensão dessa técnica. Ao combinar os multiplicadores de Lagrange com termos de penalidade quadrática, o ALM busca superar limitações numéricas comuns dos métodos de penalidade puros, promovendo maior estabilidade e garantindo convergência sob condições mais brandas (BIRGIN; MARTÍNEZ, 2014; BURACHIK; KAYA, 2012).

No contexto de funções objetivo fracionárias, como a proposta do Desafio SBPO 2025 — que visa maximizar a razão entre unidades coletadas e corredores percorridos — o ALM surge como uma alternativa promissora às transformações clássicas, como Charnes-Cooper e o algoritmo de Dinkelbach. Quando a linearização direta da função fracionária gera modelos excessivamente grandes ou instáveis, o ALM pode oferecer uma abordagem mais robusta e modular.

Entretanto, aplicar o ALM em modelos de Programação Inteira ou Mista (MILP) exige que os termos quadráticos de penalidade sejam linearizados para preservar a tratabilidade do modelo. Nesse sentido, técnicas de linearização por partes (*piecewise linearization*) são particularmente úteis. Ao aproximar funções não lineares com segmentos lineares, essas técnicas permitem capturar o comportamento penalizante do ALM sem comprometer o desempenho dos solucionadores MILP (SHERALI; FRATICELLI, 2001; WOLSEY, Laurence A., 1999). Aplicações recentes em logística demonstram que essa combinação híbrida — ALM com linearização por partes — atinge um equilíbrio interessante entre qualidade da solução e eficiência computacional (POCINHO, D., 2020).

Portanto, a proposta de empregar o Método do Lagrangeano Aumentado com técnicas de linearização por partes em modelos MILP com função objetivo fracionária parece sugerir uma contribuição original e promissora. Não sabemos se conseguiremos, mas essa abordagem não apenas ampliaria o ferramental disponível para o WOP, como também abriria caminho para modelos mais estáveis e precisos em contextos logísticos complexos.

6.7 Pré-processamentos. As jóias da coroa.

Outro aspecto crucial para a eficácia de métodos exatos é a adoção de estratégias de pré-processamento. Práticas como a fixação de variáveis com base em critérios lógicos, a remoção de restrições redundantes e a introdução de cortes válidos (ou desigualdades fortificantes) desempenham papel essencial na redução do espaço de busca e no aumento da eficiência dos solvers. A eficácia dessas técnicas é amplamente reconhecida, tanto na literatura clássica (NEMHAUSER; WOLSEY, 1988; WOLSEY, Laurence A, 1998; ACHTERBERG, 2007), quanto em estudos mais recentes sobre otimização baseada em restrições e presolve (ACHTERBERG et al., 2020), além de ter sido confirmada empiricamente na experiência relatada neste trabalho.

De forma integrada, os componentes fundamentais desta pesquisa — formulações exatas bem elaboradas, estratégias precisas de linearização, aceleração computacional via GPU e um pré-processamento inteligente — constituem o núcleo da abordagem proposta para o problema de WOP. Embora essa linha metodológica contrarie a preferência predominante por heurísticas e meta-heurísticas, os resultados obtidos demonstram que uma abordagem exata, quando cuidadosamente arquitetada, pode ser não apenas viável, mas competitiva em instâncias de porte realista.

6.8 Implicações para o Desafio SBPO 2025

A formulação fracionária proposta para o Desafio SBPO 2025 representa uma dificuldade adicional em relação aos modelos tradicionais de Order Batching and Picking (OBP). Ao adotar uma função objetivo que maximiza a razão entre o número de unidades coletadas e o número de corredores visitados, o problema adquire características pouco exploradas na literatura, exigindo abordagens mais refinadas de modelagem e solução.

Nesse cenário, a aplicação do Método do Lagrangeano Aumentado (ALM) com técnicas de linearização por partes surge como uma alternativa promissora às formulações tradicionais baseadas em constantes do tipo Big-M, como já mencionado. Além de mitigar os conhecidos problemas de instabilidade numérica associados a escolhas arbitrárias de parâmetros, o ALM permite incorporar restrições complexas de forma mais robusta, favorecendo a convergência dos algoritmos de resolução.

Especificamente no contexto do desafio, a decomposição natural do problema em subproblemas — como a seleção de pedidos e a alocação de corredores — alinha-se perfeitamente com a estrutura dual explorada pelos métodos Lagrangeanos. Isso permite tratar separadamente os componentes complicadores da formulação, como as restrições de capacidade e balanceamento entre ondas, viabilizando soluções mais eficientes do ponto de vista computacional.

Adicionalmente, a atualização dinâmica dos multiplicadores no ALM elimina a necessidade de parametrização manual de grandes constantes, uma prática comum em modelagens que utilizam o método Big-M[1]. Essa característica não apenas melhora a estabilidade numérica, mas também reduz a dependência do modelo a escolhas heurísticas de parâmetros, contribuindo para resultados mais confiáveis e replicáveis.

Em suma, a integração de uma formulação fracionária com métodos Lagrangeanos avançados posiciona esta proposta como uma contribuição metodológica relevante para o campo da otimização logística e, em particular, para o problema de Wave Order Picking no contexto do Desafio SBPO 2025.

6.9 Aplicabilidade Prática, Desafios Computacionais e Perspectivas Futuras

A popularização do comércio eletrônico e a crescente demanda por entregas rápidas e eficientes tornaram a otimização de processos logísticos em armazéns um fator crítico de competitividade. O problema proposto no Desafio SBPO 2025, ao focar na maximização da razão entre unidades coletadas e corredores visitados, representa uma formulação realista e de grande relevância prática, refletindo preocupações com eficiência operacional, redução de deslocamentos desnecessários e sustentabilidade.

O caráter combinatório do problema — que envolve decisões discretas como a seleção de pedidos e a alocação de rotas — impõe desafios computacionais significativos. A modelagem por MILP (Programação Linear Inteira Mista) e o uso de técnicas fracionárias avançadas colocam essa classe de problemas na fronteira da pesquisa em otimização logística. Sendo parte do Primeiro Desafio Mercado Livre de Otimização, o problema destaca como avanços em pesquisa

[1] O método Big-M, amplamente utilizado em formulações de Programação Linear Inteira Mista (MILP), introduz constantes suficientemente grandes para ativar ou desativar restrições condicionais via variáveis binárias. Embora facilite a modelagem, a escolha inadequada desses parâmetros pode provocar instabilidades numéricas e comprometer a eficiência dos solucionadores. Para detalhes, ver (MIT OPENCOURSEWARE, 2013), (COLUMBIA UNIVERSITY, n.d.).

operacional podem ter impacto direto na performance de sistemas logísticos modernos, reduzindo custos e melhorando o uso de recursos.

O desempenho de solucionadores MILP tem sido alavancado por inovações significativas, tanto em algoritmos quanto em infraestrutura de hardware. Solvers como o IBM CPLEX evoluíram para incorporar paralelismo via multi-threading em CPU e, mais recentemente, aceleração por GPU. Em particular, o método da barreira (*interior point*) — utilizado na resolução de relaxações lineares — pode ser executado de forma eficiente em GPU com suporte CUDA, via parâmetros como `CPXPARAM_LPMethod=4` (IBM CORPORATION, 2024). Essa abordagem aproveita o paralelismo massivo de operações matriciais, tornando-se especialmente eficaz em problemas de grande porte, como os encontrados no WOP.

Soluções nativas para GPU, como o solver NVIDIA cuOpt (NVIDIA CORPORATION, 2024), têm demonstrado excelente desempenho em problemas de roteamento e logística em larga escala. A integração dessas ferramentas com ambientes em nuvem — como os disponibilizados pela Oracle Cloud Infrastructure — viabiliza ainda mais a aplicação prática de modelos sofisticados com escalabilidade. Em paralelo, métodos de primeira ordem, como o *Primal-Dual Hybrid Gradient* (PDLP) (APPLEGATE et al., 2021), emergem como alternativas eficientes, sobretudo em problemas grandes e esparsos.

A aplicabilidade desses modelos se fortalece quando integrados a sistemas WMS (Warehouse Management Systems) com suporte a decisão em tempo real. Soluções comerciais baseadas em MILP, como as implementadas nos pacotes SAP EWM e Manhattan WMS (SAP SE, 2023; MANHATTAN ASSOCIATES, 2024), ilustram como a modelagem matemática se articula com aplicações corporativas, conectando pesquisa teórica a benefícios tangíveis em operações de armazém. Futuras pesquisas podem explorar a integração completa de métodos fracionários e técnicas de aceleração por GPU nesses ambientes, ampliando o escopo e a eficiência das soluções logísticas baseadas em otimização.

Apesar dos avanços expressivos relatados na literatura, permanecem desafios relevantes na aplicação prática de técnicas exatas ao WOP. Entre as principais limitações, destacam-se a escalabilidade dos modelos MILP frente a instâncias reais de grande porte, a sensibilidade a parâmetros de penalização em métodos de relaxação e o custo computacional associado à geração de cortes e pré-processamentos sofisticados. Embora o uso de GPUs e ambientes em nuvem mitigue parcialmente essas dificuldades, o aumento da complexidade algorítmica e a dependência de sintonia fina ainda limitam a aplicabilidade em ambientes operacionais dinâmicos (WOLSEY, Laurence A., 1999).

A crescente disponibilidade de plataformas de computação paralela e de dados operacionais em tempo real — impulsionada pela Indústria 4.0 — abre espaço para abordagens adaptativas e híbridas. Modelos com aprendizado contínuo dos parâmetros e ajustes dinâmicos de estratégia podem elevar o desempenho dos métodos exatos em cenários reais. Nesse contexto, uma linha promissora é a integração entre aprendizado de máquina e otimização exata, como nas abordagens de *learning-augmented optimization* (MIT, 2020), onde predições estatísticas informam decisões estruturais ou heurísticas iniciais, promovendo uma nova geração de soluções prescritivas para o OBP e o WOP.

A revisão aqui apresentada evidencia um panorama multifacetado do WOP, que combina desafios teóricos relevantes com alta aplicabilidade prática. A evolução das formulações, das heurísticas clássicas aos métodos Lagrangeanos Aumentados e algoritmos otimizados por GPU, revela um campo em constante transformação. Contudo, observa-se uma lacuna significativa na aplicação e avaliação de técnicas de programação fracionária mista em contextos logísticos reais, como os propostos no Desafio SBPO 2025. Este trabalho busca contribuir para o preenchimento dessa lacuna, por meio do desenvolvimento, implementação e análise de modelos híbridos que conciliem qualidade de solução, escalabilidade e robustez numérica.

7 Formulação Matemática do Problema

Como discutido nas seções anteriores, o *Wave Order Picking* (WOP) é uma estratégia logística voltada à separação eficiente de pedidos em centros de distribuição. Seu objetivo, resumindo, é agrupar pedidos que compartilham características — como itens ou corredores — e coletá-los em uma mesma "onda", reduzindo deslocamentos e otimizando o uso dos recursos operacionais. Ao contrário do foco tradicional em minimizar o *makespan* ou o tempo total de coleta, este trabalho adota uma métrica alternativa: a maximização da eficiência de coleta, medida pela razão entre o número total de unidades coletadas e o número de corredores efetivamente visitados.

Essa razão — unidades por corredor — representa um indicador direto da "densidade produtiva" do processo de picking. Ou seja, indica o quanto conseguimos "extrair" de cada corredor visitado, algo particularmente relevante em operações com espaço físico limitado ou rotatividade elevada.

Para entender melhor a lógica da formulação, imagine um armazém com corredores estreitos, nos quais a movimentação de coletores é cara (em tempo ou esforço físico). Visitar um corredor exige deslocamento, planejamento e, eventualmente, reconfiguração da rota. Portanto, quanto mais unidades conseguimos coletar por corredor visitado, melhor.

7.1 Exemplo do Desafio SBPO 2025

A instância de exemplo fornecida pelo Desafio SBPO 2025 é composta por:

- 5 pedidos, cada um solicitando diferentes quantidades de até 5 itens distintos.
- 5 corredores, cada um armazenando combinações distintas desses itens.
- Limites de tamanho da wave: mínimo de 5 unidades, máximo de 12.

Considere a seguinte solução proposta no exemplo oficial:

> Selecionar os pedidos 0, 1, 2 e 4 e visitar os corredores 1 e 3. Total de unidades coletadas: 10. Total de corredores visitados: 2. Valor objetivo: $10/2 = 5$ unidades por corredor.

Essa solução é considerada ótima para a instância fornecida, pois atinge o maior valor de eficiência (razão unidades/corredores), respeitando todos os limites e exigências de estoque. Com o exemplo em mente, e entendida a intuição por trás do modelo, apresentamos a seguir a formulação matemática completa do problema.

7.2 Notação Formal

- $O = \{0, \ldots, N_P - 1\}$: conjunto de pedidos.
- $A = \{0, \ldots, N_A - 1\}$: conjunto de corredores.
- I: conjunto de tipos de itens (SKUs).
- U_{oi}: unidades do item $i \in I$ requeridas pelo pedido $o \in O$.
- AV_{ai}: unidades disponíveis do item i no corredor a.
- $S_o = \sum_{i \in I} U_{oi}$: número total de unidades no pedido o.
- $x_o \in \{0, 1\}$: variável binária; 1 se o pedido o é incluído na wave.
- $y_a \in \{0, 1\}$: variável binária; 1 se o corredor a é visitado.

7.3 Função Objetivo

No desafio, a função objetivo [2] é apresentada na seguinte forma (SOUZA, 2025; PESQUISA OPERACIONAL, 2025):

$$Z = \frac{\sum_{o \in O} S_o \cdot x_o}{\sum_{a \in A} y_a} \tag{1}$$

No entanto, para evitar problemas computacionais — como a possibilidade de divisão por zero — e aumentar a robustez matemática [3]da formulação, optamos por modelar a função com a adição de uma constante 1 no denominador:

$$Z = \frac{\sum_{o \in O} S_o \cdot x_o}{1 + \sum_{a \in A} y_a} \tag{2}$$

A adoção de uma constante positiva no denominador da função objetivo, representada por "+1", embora pareça uma escolha técnica sutil, cumpre um papel fundamental na robustez da formulação. Em instâncias pequenas ou altamente restritivas, é possível que nenhuma seleção de corredores seja viável, o que resultaria em um denominador nulo. A inclusão dessa constante previne a divisão por zero, garantindo que o modelo permaneça bem definido mesmo nos piores cenários. Além disso, essa pequena regularização suaviza variações abruptas na razão, especialmente relevantes quando se trabalha com variáveis inteiras e decisões binárias.

Sem esse ajuste, pequenas alterações na quantidade de corredores poderiam gerar grandes oscilações no valor da função objetivo, dificultando a convergência de métodos exatos e a interpretação das soluções. Cabe ressaltar, no entanto, que o sistema de avaliação oficial do Desafio SBPO 2025 adota a forma direta da razão, utilizando $\sum_a y_a$ como denominador, sem qualquer constante adicional — o que equivale a considerar $C = 0.1$ nos termos originais do desafio (MERCADO LIVRE, 2025). Diante disso, optamos por adotar a forma regularizada durante a modelagem e o processo de linearização, a fim de garantir estabilidade computacional, mas, na

[2] A função objetivo é o coração do modelo de otimização: é ela quem diz, de forma matemática, o que significa "ir bem" ou "ir mal" em um determinado contexto. No nosso caso, ela busca maximizar a eficiência da operação — entendida como a razão entre o total de unidades coletadas e o número de corredores visitados. Isso é mais do que uma fórmula: é uma decisão estratégica sobre o que se quer otimizar de fato. Uma formulação mal pensada aqui pode levar o modelo a buscar soluções "otimamente erradas" — isto é, matematicamente válidas, mas operacionais ruins. Por isso, mais do que representar um objetivo, a função objetivo expressa a própria filosofia de desempenho do problema (IBM CORPORATION, 2023; PUC-RIO, DEPARTAMENTO DE ENGENHARIA INDUSTRIAL, 2011).

[3] Uma formulação matemática sólida precisa ir além da correção estrutural: ela deve ser capaz de se comportar bem mesmo quando os dados não colaboram. Isso é o que chamamos de *robustez numérica* — a capacidade de o modelo (e dos métodos que o resolvem) de permanecer estável, confiável e computacionalmente viável, mesmo diante de condições adversas, como variáveis próximas de zero, parâmetros mal estimados ou perturbações pequenas nos dados (BERTSIMAS; SIM, 2011; BEN-TAL; EL GHAOUI; NEMIROVSKI, 2009). Na prática, isso significa que precisamos nos antecipar a situações que podem fazer os algoritmos "travar". Uma situação clássica: variáveis que aparecem no denominador de uma fração e que podem valer zero. Esse tipo de detalhe, se não for tratado, pode comprometer não só a viabilidade da solução, mas também a estabilidade dos métodos numéricos envolvidos (NOCEDAL; WRIGHT, 2006). Para evitar isso, é comum — e necessário — adicionar pequenas constantes positivas nos denominadores, como $\epsilon = 10^{-6}$, garantindo que divisões por zero (ou valores quase nulos) não ocorram. Essas medidas simples ajudam a manter o modelo funcional e a evitar o que se conhece como *instabilidade numérica*, um dos maiores vilões em ambientes de otimização aplicada (BERTSIMAS; SIM, 2004).Além disso, quando se trabalha com dados incertos ou sujeitos a variações (como é comum em logística, por exemplo), podemos recorrer à chamada *otimização robusta* — uma área da otimização que se dedica a formular modelos cujas soluções funcionem bem não só no "mundo ideal", mas também quando os dados se desviam um pouco do esperado (BEN-TAL; EL GHAOUI; NEMIROVSKI, 2009; BERTSIMAS; SIM, 2011). No nosso contexto, essas considerações são ainda mais relevantes por duas razões: (i) lidamos com funções fracionárias que já são numericamente mais delicadas, e (ii) buscamos um modelo aplicável em cenários reais, como armazéns industriais, onde imperfeições nos dados são a regra, não a exceção. Portanto, garantir robustez numérica desde a formulação é um passo essencial — não um detalhe técnico, mas um pré-requisito prático.

etapa final de avaliação das soluções, alinhamos a métrica à especificação oficial do desafio, para permitir comparabilidade direta dos resultados.

Agora, vamos entender o que essa função realmente significa, e por que ela é coerente com o contexto logístico em que estamos inseridos. De forma simplificada, o que queremos saber é: *"Quantas unidades conseguimos coletar, em média, por corredor que precisamos visitar?"* Essa métrica — **a densidade de coleta por corredor** — para pararmos de repetir **razão...** é uma forma de medir a eficiência espacial da operação e responder isso, e a estrutura da função é a seguinte:

- **Numerador:** $\sum_{o \in O} S_o \cdot x_o$ representa o número total de unidades de itens coletadas. Aqui, $x_o = 1$ indica que o pedido o foi incluído na wave, e S_o é a quantidade total de itens solicitada nesse pedido.
- **Denominador:** $1 + \sum_{a \in A} y_a$ indica o número de corredores visitados, com o "+1" garantindo robustez, como já discutido. Cada $y_a = 1$ significa que o corredor a foi efetivamente percorrido por um coletor.

Exemplo:

Para ilustrar o funcionamento do modelo, considere a seguinte instância de entrada (baseada na `t/0001.txt` do desafio (MERCADO LIVRE, 2025)), que descreve um cenário com 5 pedidos, 5 tipos de itens distintos e 5 corredores:

```
5 5 5
2 0 3 2 1
2 1 1 3 1
2 2 1 4 2
4 0 1 2 2 3 1 4 1
1 1 1
4 0 2 1 1 2 1 4 1
4 0 2 1 1 2 2 4 1
3 1 2 3 1 4 2
4 0 2 1 1 3 1 4 1
4 1 1 2 2 3 1 4 2
5 12
```

Essa entrada descreve os seguintes dados:

- `5 5 5` indica que há 5 pedidos, 5 itens distintos (SKUs) e 5 corredores.
- As próximas 5 linhas listam os pedidos, cada um com os itens solicitados e suas quantidades.
- Em seguida, 5 linhas descrevem o conteúdo de cada corredor: quais itens ele armazena e em que quantidade.
- Por fim, a última linha define os limites inferior (5) e superior (12) para a quantidade total de unidades na wave.

Com base nessa entrada, uma solução considerada ótima seria:

- Pedidos selecionados: 0, 1, 2, 4
- Corredores visitados: 1 e 3

- Total de unidades: $S_0 = 4$, $S_1 = 2$, $S_2 = 3$, $S_4 = 1$, somando 10 unidades.
- Total de corredores: 2 (portanto, denominador $= 1 + 2 = 3$)

$$Z = \frac{10}{1+2} = \frac{10}{3} \approx 3{,}33$$

Já uma solução alternativa que coleta o máximo permitido de 12 unidades, mas exige visitar cinco corredores, resultaria em:

$$Z = \frac{12}{1+5} = \frac{12}{6} = 2{,}0$$

Esse contraste ilustra um ponto essencial: aumentar a quantidade total coletada nem sempre implica em maior eficiência. A métrica utilizada neste trabalho — a razão entre unidades coletadas e corredores visitados, com um termo constante no denominador — privilegia soluções que otimizam o uso da infraestrutura logística. Em termos práticos, ela penaliza deslocamentos excessivos que geram ganhos marginais, refletindo o custo real do esforço físico no processo de picking. A introdução do termo +1 no denominador não visa apenas evitar divisão por zero (problema potencial em instâncias com nenhuma seleção de corredores), mas também funciona como uma regularização leve que garante a definição da função objetivo mesmo em casos extremos e reforça a penalização progressiva pelo uso adicional de corredores. Esse ajuste contribui para a estabilidade do modelo, como já mencionamos, bem como viabiliza sua resolução via técnicas de programação inteira.

Entretanto, vale destacar que, embora utilizemos essa forma transformada para fins de modelagem, a avaliação final das soluções — conforme os critérios do Desafio SBPO 2025 — é feita com base na razão direta $\sum_d x_d$, ou seja, com $C = 0$. A maximização da função com +1 é, no entanto, matematicamente equivalente à da função original no sentido de que ambas produzem as mesmas soluções ótimas. Em outras palavras, a função objetivo não busca simplesmente maximizar o total de unidades nem minimizar os corredores de forma isolada. Ela equilibra os dois aspectos: é vantajoso coletar mais apenas se isso não exigir um número desproporcional de corredores, e reduzir corredores só é justificável se a perda de unidades for pequena. Essa abordagem reflete uma filosofia logística realista: *não basta coletar muito — é preciso fazê-lo de forma eficiente, minimizando o esforço envolvido.* Assim, visitar um corredor só se justifica se ele realmente "valer a viagem" (CK-12 FOUNDATION, n.d. IBM CORPORATION, 2023; SENIOR SISTEMAS, 2023).

Essa formulação não apenas espelha de forma mais fiel a dinâmica operacional do problema *Wave Order Picking*, como também sustenta as técnicas de otimização exata que serão detalhadas nas seções seguintes deste artigo. Entretanto, para que a função objetivo atue de forma coerente com a realidade operacional modelada, é necessário que seja acompanhada por um conjunto robusto de restrições. São essas restrições que delimitam o espaço de soluções viáveis, traduzindo requisitos logísticos fundamentais — como a quantidade mínima e máxima de unidades por wave, a disponibilidade de estoque e a estrutura física do armazém — em expressões matemáticas que o modelo deve respeitar. É justamente a combinação entre uma função objetivo realista e restrições bem formuladas que confere poder expressivo ao modelo.

7.4 Restrições do Modelo

A formulação matemática, como abordado no Desafio SBPO 2025 (MERCADO LIVRE, 2025), impõe um conjunto específico de restrições que visam garantir a viabilidade logística e operacional das soluções geradas. Essas restrições traduzem, em linguagem matemática, as exigências práticas da tarefa: selecionar uma combinação eficiente de pedidos (a "wave") e de corredores a serem visitados, assegurando tanto a cobertura da demanda quanto o respeito aos limites de capacidade do sistema. A seguir, descrevemos cada uma delas

7.4.1 Limite Inferior de Unidades na Wave

Esta restrição garante que a quantidade total de unidades coletadas na wave atinja um patamar mínimo previamente definido:

$$\sum_{o \in O} S_o \cdot x_o \geq \text{LB} \tag{3}$$

onde:

- S_o: total de unidades do pedido o;
- $x_o \in \{0, 1\}$: variável que indica se o pedido o foi selecionado;
- LB: limite inferior de unidades na wave.

Esta restrição corresponde à formulação alternativa presente na literatura:

$$\sum_{o \in O'} \sum_{i \in I_o} u_{oi} \geq \text{LB}$$

sendo $S_o = \sum_{i \in I_o} u_{oi}$ e $x_o = 1$ se $o \in O'$.

7.4.2 Limite Superior de Unidades na Wave

De forma análoga, esta restrição impõe um teto para o total de unidades na wave:

$$\sum_{o \in O} S_o \cdot x_o \leq \text{UB} \tag{4}$$

onde UB representa o limite superior da capacidade da wave.

7.4.3 Restrição de Cobertura de Itens (Capacidade/Estoque)

Para cada item $i \in I$, a quantidade total requisitada nos pedidos selecionados deve estar contida na soma dos estoques disponíveis nos corredores escolhidos:

$$\sum_{o \in O} U_{oi} \cdot x_o \leq \sum_{a \in A} AV_{ai} \cdot y_a \quad \forall i \in I \tag{5}$$

com:

- U_{oi}: unidades do item i no pedido o;
- AV_{ai}: unidades do item i disponíveis no corredor a;
- $y_a \in \{0, 1\}$: variável que indica se o corredor a será visitado.

Esta restrição garante que a demanda da wave possa ser totalmente satisfeita com os itens armazenados nos corredores selecionados.

Ao examinarmos a formulação matemática desenvolvida até aqui, não é difícil traçar paralelos com problemas clássicos da teoria da otimização combinatória, como o Problema da Mochila (*Knapsack Problem*) e o Problema da Cobertura por Conjuntos (*Set Covering Problem*). A estrutura do *Wave Order Picking* envolve justamente a seleção simultânea de pedidos e corredores — um entrelaçamento de decisões que remete à maximização do aproveitamento de recursos sob restrições de capacidade, à garantia de cobertura mínima com o menor custo logístico possível, e a outros dilemas típicos de formulações NP-difíceis (WIKIPÉDIA, 2010; EUROCC NCC BELGIUM, s.d.; DCC/UFMG, s.d.). Apesar dessa complexidade, é justamente esse cenário desafiador que nos motiva a explorar abordagens metodológicas rigorosas e sistemáticas, servindo como ponto de partida para o próximo passo desta investigação: a busca por soluções exatas, implementáveis e reproduzíveis.

7.5 Processo de Linearização da Função Objetivo

A função objetivo do problema de Wave Order Picking (WOP), conforme definida no Desafio SBPO 2025, apresenta-se na forma de uma razão entre expressões lineares. Essa formulação, que busca maximizar a eficiência de coleta (unidades por corredor), é um dos **pilares centrais** da nossa abordagem, mas sua natureza fracionária a torna intrinsecamente não linear, o que impõe um desafio significativo para a resolução por métodos tradicionais de Programação Linear Inteira (PLI).[4] Para superar essa limitação e tornar o problema tratável numericamente por *solvers* de PLI, diversas técnicas de linearização foram desenvolvidas. Neste subcapítulo, exploraremos as **três principais estratégias matemáticas** que avaliamos e comparamos para a linearização da nossa função objetivo: o método da variável inversa, a transformação de Charnes-Cooper e o algoritmo iterativo de Dinkelbach. As seções a seguir detalharão os fundamentos teóricos, os procedimentos de implementação e as implicações práticas de cada uma dessas abordagens no contexto do WOP.

7.5.1 A Variável Inversa

Uma forma comum de lidar com funções objetivo fracionárias como a nossa, do tipo max $Z = \frac{N(x)}{D(y)}$, em que $N(x)$ é o numerador e $D(y)$ o denominador, é adicionar variáveis e restrições para transformar a razão em um produto equivalente. Essa técnica é baseada em métodos clássicos como os de Charnes-Cooper (CHARNES; COOPER, 1962) e Dinkelbach (DINKELBACH, 1967). No contexto do nosso problema, onde:

$$Z = \frac{\sum_{o \in O} S_o \cdot x_o}{1 + \sum_{a \in A} y_a}$$

sendo, como já sabemos, $\sum_{o \in O} S_o \cdot x_o$ o número total de unidades coletadas $(N(x))$ e $1 + \sum_{a \in A} y_a$ o número de corredores visitados, com aquela constante de robustez[5], $(D(y))$, o cerne dessa técnica reside justamente nessa transformação razão-produto mencionada, de tal modo que as variáveis e restrições garantam equivalência com o problema original (CHARNES; COOPER, 1962; DINKELBACH, 1967; WOLSEY, Laurence A., 1999). Os passos são os seguintes:

1. Introduzir uma nova variável contínua z que represente o inverso do denominador: $z = 1/D(y) = 1/\left(1 + \sum_{a \in A} y_a\right)$.
2. Transformar a maximização da *razão* $N(x)/D(y)$ em maximização do *produto*: $N(x) \cdot z = \left(\sum_{o \in O} S_o \cdot x_o\right) \cdot z$.

[4] A presença de quocientes em funções objetivo fracionárias impede a aplicação direta dos algoritmos de otimização linear, que são projetados para problemas com relações lineares entre variáveis.

[5] Vide nota 3

3. Linearizar o produto das variáveis $x_o \cdot z$ (onde x_o é binária e z é contínua) introduzindo uma variável nova $w_o = x_o \cdot z$ para cada pedido o.

O resultado esperado dessa abordagem é a função objetivo linearizada: $\mathbf{max} \sum_{o \in O} \boldsymbol{S_o \cdot w_o}$.

Para ilustrar a simplicidade conceitual dessa técnica, considere o exemplo:

$$\begin{aligned} \max \quad & \frac{2x}{y} \\ \text{sujeito a} \quad & 1 \leq x \leq 5 \\ & 1 \leq y \leq 3 \end{aligned}$$

Como a ideia é **maximizar a razão entre dois valores**, os $2x$ e o y, respeitando as restrições de que o primeiro deve estar entre 1 e 5, e o segundo entre 1 e 3, a **estratégia de modelagem** adotada é "virar" o y de cabeça para baixo. Matematicamente, "virar" significa usar o *inverso*. Por exemplo, o inverso de 2 é $\frac{1}{2}$, o inverso de 4 é $\frac{1}{4}$ e o inverso de y é $\frac{1}{y}$. A partir disso, criamos uma nova variável para representar esse inverso. Vamos chamá-la de z. Por conta disso, então, dizemos que $z = \frac{1}{y}$. Agora, observe como essa substituição é poderosa: se pegarmos a equação original e substituirmos $\frac{1}{y}$ por z, transformamos uma divisão em uma multiplicação, o que torna o problema mais simples de manipular matematicamente.

$$\frac{2x}{y} = 2x \cdot \frac{1}{y} = 2x \cdot z$$

Ou seja, **a divisão sumiu**. Nosso problema virou simplesmente: max $2x \cdot z$. Mas é importante lembrar que z não é uma variável qualquer: ele depende diretamente de y. Por isso, precisamos **adicionar uma restrição** que torne essa dependência explícita no modelo. Como essa restrição é bem direta: impomos que $y \cdot z = 1$. Com isso, o modelo "entende"que z tem que ser exatamente o inverso de y, e não outro valor qualquer. Essa equação é o "contrato" que garante que, se definirmos $z = \frac{1}{y}$, então $2x \cdot z$ será exatamente igual a $\frac{2x}{y}$. Assim, a transformação mantém a lógica da razão original, mas de forma mais amigável à modelagem matemática e computacional. Resultado:

$$\begin{aligned} \max \quad & 2x \cdot z \\ \text{sujeito a} \quad & 1 \leq x \leq 5 \\ & 1 \leq y \leq 3 \\ & y \cdot z = 1 \end{aligned}$$

Veja o algoritmo abaixo, que formaliza essa abordagem de linearização aplicável ao nosso problema:

Algoritmo 1 Linearização via Variável Inversa aplicada ao WOP

Require: Conjuntos de pedidos $\mathcal{O}$, corredores $\mathcal{A}$, unidades S_o, tolerância ε
Ensure: Solução ótima (x^*, y^*, z^*, w^*)
1: Introduza a variável contínua $z > 0$
2: Para cada pedido $o \in \mathcal{O}$, defina $w_o = x_o \cdot z$
3: Imponha a equação de normalização:

$$z \cdot \left(1 + \sum_{a \in \mathcal{A}} y_a\right) = 1$$

4: Linearize $w_o = x_o \cdot z$ com as restrições:

$$\begin{cases} w_o \leq M \cdot x_o \\ w_o \leq z \\ w_o \geq z - M(1 - x_o) \\ w_o \geq 0 \end{cases} \qquad \forall o \in \mathcal{O}$$

5: Maximize:

$$\sum_{o \in \mathcal{O}} S_o \cdot w_o$$

6: Sujeito às restrições do modelo original

Para que você entenda melhor o que estamos tentando explicar, imagine que está dentro do armazém do Mercado Livre, que - ao menos para este exemplo, deve ser grande, tentando decidir quais pedidos vale a pena coletar e por quais corredores precisa passar para isso. Eis o desafio: coletar o máximo possível, mas andando o mínimo necessário. No entanto, em vez de pensar em “quantos itens por corredor” você consegue obter, você decide enxergar o problema sob uma lente invertida. Se tiver sorte, talvez você pense: “E se eu puder transformar o número de corredores visitados em um fator que *diminui* os pedidos selecionados?” — como se todo pedido escolhido passasse por um redutor de eficiência proporcional ao esforço exigido.

Se você pensa assim, parabéns, você é um gênio! Ou, pelo menos, está no caminho certo para entender essa técnica melhor do que nós! Mas voltando ao que interessa, essa ideia de “redutor” é exatamente o que a variável inversa faz, e é representado pela variável contínua z, que corresponde ao inverso da quantidade de corredores visitados (com um ajuste de robustez: $z = \frac{1}{1+\sum_{a \in A} y_a}$). O algoritmo da variável inversa funciona assim: ele associa a cada pedido uma nova variável auxiliar $w_o = x_o \cdot z$, que representa a “versão reescalada” da seleção daquele pedido, já levando em conta o esforço logístico.

Em termos práticos, é como se você perguntasse: “Vale a pena pegar esse pedido, sabendo que ele será *distorcido* pela minha eficiência atual?”. A função objetivo passa então a somar os valores $S_o \cdot w_o$ de todos os pedidos, procurando a melhor combinação que maximize esse total — mesmo considerando o esforço embutido no z. Essa mudança de perspectiva permite reformular o problema em termos lineares, desde que as multiplicações $x_o \cdot z$ sejam devidamente linearizadas por desigualdades auxiliares. Ao final, você obtém uma solução que equilibra qualidade e esforço — sem precisar resolver o problema fracionário diretamente.

Seja por sorte ou acaso, o exemplo acima nos ajuda a demonstrar como a técnica de linearização por variável inversa pode ser tão **intuitiva** quanto **poderosa** na transformação de problemas fracionários em lineares (CHARNES; COOPER, 1962). Contudo, em problemas mais complexos — onde x e y representam somatórios ou combinações não lineares de variáveis —, essa estratégia, embora válida e frequentemente essencial, pode exigir complementos teóricos para garantir a tratabilidade computacional.

7.5.2 O Método Big-M

O método Big-M é uma técnica clássica de PLI utilizada para modelar restrições condicionais e variáveis artificiais, especialmente em problemas que envolvem variáveis binárias e a necessidade de ativar ou desativar certas restrições conforme o valor dessas variáveis. Confuso, não é? Vamos esclarecer. Acima, a substituição do inverso e a transformação da razão em produto foram feitas entre variáveis contínuas, o que tornou a modelagem direta e relativamente simples. Porém, em nosso problema real, enfrentamos uma situação mais complexa: o produto a ser linearizado envolve uma variável binária x_o e uma variável contínua z, resultando na nova variável $w_o = x_o \cdot z$. Essa mistura de tipos de variáveis dificulta um pouco as coisas, já que não podemos simplesmente substituir o produto por uma variável sem garantir que w_o assuma o valor correto conforme o valor de x_o. Em outras palavras, precisamos assegurar que:

$$w_o = \begin{cases} z, & \text{se } x_o = 1 \\ 0, & \text{se } x_o = 0 \end{cases}$$

Para formalizar essa relação mantendo o modelo linear, utilizamos o *método Big-M*[6], que introduz um parâmetro M suficientemente grande (e com esse nome nem poderia ser diferente) para ativar ou desativar restrições condicionais (WOLSEY, Laurence A., 1999). Por meio dessas restrições, o modelo impõe os limites corretos para w_o dependendo do valor de x_o, garantindo a equivalência exata com o produto original. Esse método é amplamente adotado em programação linear inteira para tratar produtos entre variáveis binárias e contínuas, sendo essencial para manter a linearidade e viabilizar a resolução eficiente do problema.

Ao linearizar o produto $w_o = x_o \cdot z$, onde x_o é binária e z contínua, é necessário garantir que w_o assuma o valor correto dependendo do valor de x_o. Para isso, utiliza-se o método Big-M, que introduz um parâmetro M suficientemente grande para ativar ou desativar restrições condicionais (WOLSEY, Laurence A., 1999).

As restrições típicas para modelar $w_o = x_o \cdot z$ são:

$$\begin{cases} w_o \leq z, \\ w_o \leq M x_o, \\ w_o \geq z - M(1 - x_o), \\ w_o \geq 0, \end{cases}$$

onde M deve ser escolhido grande o suficiente para não restringir soluções viáveis, mas não tão grande a ponto de causar instabilidades numéricas ou dificultar a resolução (PEDROSO, 2024). Esse conjunto de restrições completa a transformação do problema original em um modelo linear inteiro misto compatível com os principais solvers de otimização. Em problemas ainda mais complexos, onde x e y são somatórios ou combinações não lineares, técnicas como as **desigualdades de McCormick** são indispensáveis para preservar a integridade do modelo (MCCORMICK, 1976).

7.5.3 Desigualdades de McCormick

As desigualdades de McCormick, introduzidas por Garth P. McCormick em 1976 (MCCORMICK, 1976), são uma técnica fundamental para a linearização de produtos não lineares entre variáveis contínuas e/ou binárias em problemas de Programação Linear Inteira Mista (MILP).

[6] O método Big-M é amplamente utilizado em modelagem matemática para representar restrições condicionais, variáveis artificiais no método simplex e para linearizar produtos envolvendo variáveis binárias e contínuas, sendo fundamental para a construção de modelos de Programação Linear Inteira Mista (MILP) (FACOM-UFMS, 2009; LUANDA, 2022).

Essa técnica permite substituir termos multiplicativos por um conjunto de restrições lineares que formam a envoltória convexa mais apertada possível para o produto, garantindo que a solução do modelo linearizado seja equivalente à do modelo original não linear, dentro dos limites das variáveis (WOLSEY, Laurence A., 1999; SHERALI; ADAMS, 1999).

Considere o produto $\boldsymbol{w} = \boldsymbol{x} \cdot \boldsymbol{z}$, onde x é uma variável binária ($\boldsymbol{x} \in \{\mathbf{0}, \mathbf{1}\}$) e z é uma variável contínua limitada por $\underline{\boldsymbol{z}} \leq \boldsymbol{z} \leq \overline{\boldsymbol{z}}$. As desigualdades de McCormick para esse produto são dadas por: $\boldsymbol{w} \leq \overline{\boldsymbol{z}} \cdot \boldsymbol{x}$, $\boldsymbol{w} \geq \underline{\boldsymbol{z}} \cdot \boldsymbol{x}$, $\boldsymbol{w} \leq \boldsymbol{z} - \underline{\boldsymbol{z}} \cdot (\mathbf{1} - \boldsymbol{x})$, $\boldsymbol{w} \geq \boldsymbol{z} - \overline{\boldsymbol{z}} \cdot (\mathbf{1} - \boldsymbol{x})$. Quando $\boldsymbol{x} = \mathbf{0}$, tem-se $\boldsymbol{w} = \mathbf{0}$, pois as desigualdades limitam w a zero. Quando $\boldsymbol{x} = \mathbf{1}$, tem-se $\boldsymbol{w} = \boldsymbol{z}$, pois as desigualdades forçam w a assumir o valor de z.

Dessa forma, o produto é linearizado sem perda de exatidão, dentro dos limites estabelecidos para z, o que é crucial para a modelagem precisa e eficiente de problemas que combinam decisões binárias com variáveis contínuas. No contexto do nosso problema, essa abordagem é particularmente útil, já que tanto a função objetivo quanto algumas restrições envolvem produtos desse tipo — como, por exemplo, a seleção de pedidos ou corredores (variáveis binárias) associada a tempos, custos ou inversos de somatórios (variáveis contínuas).

7.5.4 Vantagens e Desvantagens das Desigualdades de McCormick

Para facilitar a análise crítica dessa técnica de linearização, organizamos seus principais pontos positivos e negativos na Tabela **??**. Como se pode observar, embora as desigualdades de McCormick ofereçam uma forma exata e versátil de tratar produtos bilineares, elas não estão isentas de limitações práticas que exigem atenção especial durante a modelagem.

Tabela 1: Vantagens e desvantagens das desigualdades de McCormick

Vantagens	**Desvantagens**
Linearização exata dentro dos limites: fornece a envoltória convexa mais apertada possível, garantindo equivalência ao modelo original dentro dos domínios definidos (MCCORMICK, 1976).	**Aumento da complexidade estrutural:** a introdução de variáveis auxiliares e restrições adicionais pode inflar o modelo e dificultar sua resolução (WOLSEY, Laurence A., 1999).
Versatilidade: aplicável a produtos entre variáveis binárias, contínuas ou mistas, sendo útil em uma ampla gama de problemas práticos como roteirização, alocação e planejamento (WOLSEY, Laurence A., 1999).	**Dependência de limites bem definidos:** a qualidade da aproximação depende fortemente da definição precisa de z_{min} e z_{max}; valores frouxos enfraquecem a relaxação (MCCORMICK, 1976).
Melhoria da relaxação linear: fortalece o modelo contínuo relaxado, acelerando algoritmos como branch-and-bound e cortes (SHERALI; ADAMS, 1999).	**Escalabilidade limitada:** em problemas de grande porte, o número crescente de restrições pode comprometer a viabilidade computacional (SHERALI; ADAMS, 1999).
Compatibilidade com solvers comerciais: podem ser implementadas diretamente em ferramentas como Gurobi, CPLEX e CBC (COIN-OR, 2020; GUROBI OPTIMIZATION, LLC, 2023; IBM, 2017).	**Cuidados numéricos necessários:** como no método Big-M, o uso de constantes grandes pode causar instabilidades se não for bem calibrado (WOLSEY, Laurence A., 1999).

7.5.5 O Método de Charnes-Cooper

A transformação de Charnes-Cooper, desenvolvida por Abraham Charnes e William W. Cooper (CHARNES; COOPER, 1962), é considerada, na literatura, uma das abordagens mais elegantes e matematicamente rigorosas para converter problemas de PLF em PL equivalentes (MAXWELL, 2014; DUARTE, 2019). Essa técnica não apenas solucionou o desafio clássico da área[7], mas também abriu caminho para toda uma classe de métodos de linearização de funções objetivo fracionárias (CHARNES; COOPER, 1962; BOUERI; ROCHA; RODOPOULOS, 2019; ALMEIDA; REBELATTO, 2019). No contexto do nosso estudo, ela merece destaque especial devido à sua efetividade e aplicabilidade direta. Por essa razão, apresentamos a seguir alguns detalhes da sua aplicação ao nosso caso de estudo.

O cerne da transformação de Charnes-Cooper está na introdução de uma variável contínua t que, diferentemente da variável z anterior, não serve apenas como o inverso do denominador da função fracionária. Aqui a variável nova desempenha um papel maior, servindo de fator de escala que permite reescrever todo o problema em uma nova base de variáveis, de forma sistemática, homogênea e, sobretudo, linear (MARQUES JÚNIOR, 2021).

Formalmente, definimos esse fator de escala assim:

$$t = \frac{1}{D(y)} = \frac{1}{1 + \sum_{a \in \mathcal{A}} y_a}$$

A partir disso, transformamos todas as variáveis do modelo original. Em termos práticos, significa multiplicá-las também por t, obtendo, por conseguinte, um conjunto de variáveis escaladas:

$$\begin{aligned} x'_o &= t \cdot x_o, \quad \forall o \in \mathcal{O} \\ y'_a &= t \cdot y_a, \quad \forall a \in \mathcal{A} \end{aligned}$$

Essa transformação tem uma vantagem prática fundamental: é válida sempre que o denominador da função fracionária original for estritamente positivo. No nosso caso, isso está garantido graças à inclusão daquela constante "+1" robustez[8] no denominador da função objetivo, que assegura que $D(y) > 0$ para qualquer combinação de $y_a \in \{0, 1\}$, evitando o problemas como a divisão por zero ou indefinições que, em modelos reais, podem ser fatais para o solver (SCHAIBLE, 1981).

Aplicando essa transformação ao nosso problema, conforme descrito originalmente em (CHARNES; COOPER, 1962; MAXWELL, 2014), cuja função objetivo era:

$$\max Z = \frac{\sum_{o \in \mathcal{O}} S_o \cdot x_o}{1 + \sum_{a \in \mathcal{A}} y_a}$$

[7] O principal desafio clássico que a transformação de Charnes-Cooper resolveu está relacionado à natureza não linear dos problemas de Programação Linear Fracionária (PLF). Nestes problemas, a função objetivo é uma razão entre duas funções lineares, o que impede a aplicação direta dos métodos tradicionais de Programação Linear (PL), pois a presença da fração torna o problema não linear e, portanto, mais complexo de resolver.

A transformação proposta por Charnes e Cooper (CHARNES; COOPER, 1962) consiste em uma mudança de variáveis que fixa o denominador da função objetivo igual a 1, convertendo o problema original em um problema de programação linear equivalente. Dessa forma, o problema fracionário passa a ser tratado com as mesmas técnicas rigorosas e eficientes da programação linear clássica, superando a dificuldade imposta pela não linearidade da função objetivo.

Esse avanço foi fundamental para diversas áreas, como a Análise Envoltória de Dados (DEA), onde o modelo CCR utiliza essa transformação para avaliar a eficiência relativa de unidades produtivas, linearizando a função objetivo que originalmente é uma razão entre outputs e inputs (MAXWELL, 2014; COELLI; RAO; BATTESE, 1998; CHARNES; COOPER; RHODES, 1978).

[8] Vide nota 3.

sujeita às restrições estabelecidas no Desafio[9], obtemos a seguinte versão transformada e linearizada:

$$\max Z' = \sum_{o \in \mathcal{O}} S_o \cdot x'_o$$

Naturalmente, mas não tão óbvio, essa nova formulação exige um novo conjunto de restrições, a começar pela chamada restrição de normalização:

$$t + \sum_{a \in \mathcal{A}} y'_a = 1$$

Essa equação garante que a equivalência entre o modelo original e o transformado seja mantida, ao assegurar que $t \cdot (1 + \sum y_a) = 1$. Isso, por sua vez, justifica toda a reparametrização feita até aqui. Além disso, todas as demais restrições do modelo original também precisam ser multiplicadas por t e reescritas no novo sistema de variáveis. A **restrição de limite inferior de unidades**, por exemplo::

$$\sum_{o \in \mathcal{O}} S_o \cdot x_o \geq LB$$

transforma-se em:

$$\sum_{o \in \mathcal{O}} S_o \cdot x'_o \geq LB \cdot t$$

sujeita[10] a:

$$\begin{aligned} x'_o &\in \{0, t\}, \quad \forall o \in \mathcal{O} \\ y'_a &\in \{0, t\}, \quad \forall a \in \mathcal{A} \\ t &> 0 \end{aligned}$$

[9]Vide subcapítulo Restrições do Modelo.

[10]Em modelos de otimização PL, PLI ou MILP, as variáveis de decisão sempre estão associadas a um **domínio**, ou seja, ao conjunto de valores possíveis que cada variável pode assumir. As *restrições de domínio das variáveis* consistem na explicitação, no modelo matemático, desses conjuntos de valores, limitando a região factível apenas aos pontos cujas variáveis satisfaçam tais condições.

Por exemplo:

- determinada variável deve ser não-negativa e inteira:

$$x_i \geq 0, \quad x_i \in \mathbb{Z}$$

- outra pode ser binária:

$$y_j \in \{0, 1\}$$

- e até as de proporção podem ser restritas a um intervalo unitário:

$$0 \leq p_k \leq 1$$

Essas restrições são normalmente explicitadas após as equações principais do modelo, como fizemos várias vezes aqui, geralmente ao final da formulação, indicando, ou tentando indicar, os limites impostos a cada uma ou a a um grupo de variáveis (SOUZA; JESUS GOMES FURTADO; HORTA, 2018). Logo, ao resolver um problema de otimização, os algoritmos levam em consideração não apenas as restrições relacionais do problema (igualdades e desigualdades), mas também os domínios declarados para cada variável. Essas últimas, aliás, são essenciais, pois impedem que soluções não factíveis — tais como quantidades negativas, frações de elementos que só podem ser inteiros ou decisões não-binárias — sejam consideradas candidatas à solução ótima. Dito como aprendemos em aula, essas restrições de domínio definem o espaço viável do problema, restringindo o conjunto de soluções admissíveis (KOLMAN, 1995; SOUZA; JESUS GOMES FURTADO; HORTA, 2018).

A restrição $t > 0$ é essencial, pois, sem ela o fator de escala se tornaria inválido e a transformação perderia sentido. Já as inclusões $x'_o, y'_a \in \{0, t\}$ refletem o fato de que as variáveis originais x_o e y_a assumem valores binários, ou seja, 0 ou 1. Esse detalhe é importante porque preserva o caráter inteiro e binário do modelo original, o que é especialmente relevante em problemas de Programação Inteira Mista com função objetivo fracionária (MAXWELL, 2014).

Esse comportamento das variáveis transformadas, no entanto, exige um cuidado adicional: embora x'_o e y'_a devam pertencer ao conjunto $\{0, t\}$, nada garante, por si só, que isso será respeitado automaticamente pelo solver. Por isso, é necessário impor condições extras no modelo — um ponto que nos leva ao tratamento das variáveis binárias após a transformação. A literatura apresenta duas abordagens clássicas para lidar com esse tipo de estrutura após a transformação:

- **Restrições adicionais:**

$$x'_o \leq t, \quad \forall o \in \mathcal{O}$$

$$x'_o \geq t - M(1 - z_o), \quad \forall o \in \mathcal{O}$$

  onde z_o é uma nova variável binária e M é uma constante suficientemente grande para garantir a equivalência.

- **Substituição direta:**

$$x'_o = z_o \cdot t$$

  Essa técnica é mais intuitiva, mas reintroduz produtos bilineares $(t \cdot z_o)$, o que nos obriga a usar outras técnicas de linearização.

Segundo Schaible e Ibaraki (SCHAIBLE; IBARAKI, 1983), a primeira abordagem tende a ser mais estável do ponto de vista numérico e oferece melhor desempenho prático, especialmente quando implementada em solvers comerciais como CPLEX ou Gurobi (IBM, 2017; GUROBI OPTIMIZATION, LLC, 2023).Por fim, uma das maiores vantagens da transformação de Charnes-Cooper é a simplicidade da reconstrução da solução original:

$$x_o = \frac{x'_o}{t}, \quad \forall o \in \mathcal{O}$$
$$y_a = \frac{y'_a}{t}, \quad \forall a \in \mathcal{A}$$

Como $x'_o, y'_a \in \{0, t\}$ e $t > 0$, essa recuperação sempre retorna valores binários, preservando a coerência com o modelo original.

Veja o algoritmo abaixo, que resume o processo de transformação acima aplicado ao nosso problema. É uma versão simplificada do original, mas mantém a essência da técnica, podendo ser facilmente adaptado para diferentes contextos e problemas fracionários, desde que as condições de normalização e linearidade sejam respeitadas.

Algoritmo 2 Transformação de Charnes-Cooper adaptada ao WOP

Require: Conjuntos $\mathcal{O}$, $\mathcal{A}$, unidades S_o, tolerância ε
Ensure: Solução ótima em variáveis escaladas (x'_o, y'_a, t)

1: Defina $t > 0$ e as variáveis escaladas:

$$x'_o = t \cdot x_o, \quad y'_a = t \cdot y_a, \quad \forall o \in \mathcal{O},\ a \in \mathcal{A}$$

2: Imponha a restrição de normalização:

$$t + \sum_{a \in \mathcal{A}} y'_a = 1$$

3: Reescreva as restrições originais, multiplicadas por t:

$$\sum_{o \in \mathcal{O}} S_o \cdot x'_o \geq LB \cdot t, \quad \ldots$$

4: Restrinja os domínios das variáveis:

$$x'_o \in [0, t], \quad y'_a \in [0, t]$$

5: Maximize:

$$\sum_{o \in \mathcal{O}} S_o \cdot x'_o$$

Vamos tentar explicar essa transformação de forma mais intuitiva usando uma analogia com o exemplo do armazém do Mercado Livre que mencionamos no início. Depois de definida a nova variável contínua t, que representa o esforço ou o "custo embutido" no denominador do problema fracionário, o algoritmo executa um reescalonamento sistemático de todas as variáveis do modelo. Isso significa que, ao invés de atuar diretamente sobre a razão, transformamos o problema em um novo ambiente, onde as variáveis escaladas já incorporam a fração original — permitindo que o modelo seja resolvido com ferramentas lineares clássicas.

Cada etapa do algoritmo tem uma interpretação intuitiva:

- Ao definir $x'_o = t \cdot x_o$ e $y'_a = t \cdot y_a$, estamos comprimindo (ou expandindo) as decisões em função do nível de esforço representado por t.

- A condição $t + \sum_{a \in \mathcal{A}} y'_a = 1$ garante que a escala escolhida seja coerente com a realidade, já que impõe uma régua única sobre todas as variáveis.

- As restrições originais são mantidas, mas agora adaptadas à nova escala. Por exemplo, a soma, para todo pedido o no conjunto $\mathcal{O}$, do número de unidades S_o multiplicado pela variável escalada x'_o, deve ser maior ou igual a LB vezes o fator de escala t

$$\sum_{o \in \mathcal{O}} S_o \cdot x'_o \geq LB \cdot t$$

  quer dizer que ainda é necessário coletar um número mínimo de unidades — mas esse mínimo agora é proporcional ao esforço total permitido.

- A função objetivo, ao fim e ao cabo, resta linear:

$$\max \sum_{o \in \mathcal{O}} S_o \cdot x'_o$$

Uma vez obtida a solução ótima do problema escalado, basta reverter o reescalonamento para recuperar as decisões originais:

$$x_o = \frac{x'_o}{t}, \quad y_a = \frac{y'_a}{t}$$

Considere:

$$\max \frac{2x}{1+y}, \quad x \in \{0,1\},\ y \in \{0,1\}$$

Defina $t = \frac{1}{1+y}$, e aplique a transformação:

$$x' = t \cdot x, \quad y' = t \cdot y, \quad t + y' = 1$$

O objetivo se torna:

$$\max 2x', \quad \text{com } x' \in \{0,t\},\ y' \in \{0,t\},\ t > 0$$

A recuperação da solução original é imediata:

$$x = \frac{x'}{t}, \quad y = \frac{y'}{t}$$

Ainda que, à primeira vista, essa transformação possa parecer sofisticada, para não dizer difícil, ela é reconhecida na literatura como uma técnica simples, como demonstrado nos trabalhos originais dos autores (CHARNES; COOPER, 1962) e nos estudos posteriores (FERREIRA, 2015). Entretanto, é importante esclarecer um ponto técnico que costuma ser mal interpretado: embora a reformulação pareça garantir que as variáveis recuperadas serão automaticamente binárias (isto é, 0 ou 1), isso nem sempre ocorre de forma natural. Para que a integridade binária das variáveis seja preservada, é necessário impor restrições adicionais ao modelo transformado — como garantir que $x'_o \in \{0, t\}$ e que o denominador seja estritamente positivo, assegurando a viabilidade da inversão e da reinterpretação das variáveis escaladas[11].

Mas talvez isso tudo ainda soe meio abstrato. O exemplo prometido pode ajudar. Imagine-se novamente dentro do armazém do Mercado Livre, tentando decidir quais pedidos vale a pena coletar e por quais corredores passar. Só que, desta vez, em vez de aplicar um redutor diretamente sobre os pedidos, como fizeste antes, você resolve redefinir a escala do problema inteiro. Você inteligentemente pensa: "E se eu medir tudo — pedidos, corredores, restrições — com base no *quanto estou disposto a me esforçar*?" Como já dissemos, você é um gêino. Esse esforço total é representado por uma variável contínua t, definida como:

$$t = \frac{1}{1 + \sum_{a \in \mathcal{A}} y_a}$$

Logo, quanto mais corredores forem visitados, menor será o valor de t, e todo o modelo será proporcionalmente reescalado — como se o esforço envolvido comprimisse as decisões possíveis. O método de Charnes-Cooper, portanto, não busca balancear numerador e denominador diretamente. Ele muda o cenário onde o problema é resolvido: transforma uma razão complicada em uma simples soma linear. É como se você dissesse: "não vou lidar com a fração — vou mudar o sistema de medidas" e, com isso, você ainda ganha diversas vantagens práticas, como a estabilidade numérica e a compatibilidade com variáveis inteiras, a facilidade de recuperação

[11] A literatura recomenda, nesse contexto, o uso de restrições de acoplamento ou técnicas como as desigualdades de McCormick para assegurar que a transformação mantenha as variáveis no conjunto binário após a divisão pelo fator de escala t. Vide (FERREIRA, 2015).

da solução original e, principalmente, uma linearização exata — sem necessidade de métodos iterativos ou aproximações não lineares.

Se você recordar bem do exemplo anterior sobre a variável inversa, vai perceber que os dois métodos compartilham uma filosofia de linearização adaptativa semelhante. A diferença, no entanto, está na forma que cada um propõe. Aquele impõe um redutor diretamente sobre os pedidos — cada x_o é multiplicado por uma variável contínua z, que representa o inverso do esforço (ou seja, do número de corredores envolvidos). Esse atua de forma mais ampla, já que, em vez de aplicar um redutor localizado, redefine o sistema de medidas de todo o problema, reconstruindo o universo decisório a partir do fator de escala global t, proporcional ao esforço total estimado, de tal modo que todas as variáveis sejam ajustadas de acordo com essa nova régua.

Grosso modo, *Enquanto a variável inversa aplica uma lente sobre os pedidos, Charnes-Cooper muda o tamanho do universo onde a decisão é tomada.* Para destacar as diferenças entre essas abordagens de forma mais objetiva, apresentamos a seguir uma comparação entre os métodos discutidos: a variável inversa, a transformação de Charnes-Cooper e, na sequência, o algoritmo iterativo de Dinkelbach.

Tabela 2: Comparação entre técnicas de linearização para problemas fracionários

Critério	**Variável Inversa**	**Charnes-Cooper**	**Dinkelbach**
Transformação	Parcial (denominador)	Completa (todas as variáveis)	Iterativa
Estabilidade Numérica	Moderada	Alta	Dependente da convergência
Variáveis Adicionais	$O(\|\mathcal{O}\|)$	$O(\|\mathcal{O}\| + \|\mathcal{A}\| + 1)$	Nenhuma (por iteração)
Compatibilidade com PLI	Requer adaptações	Direta com adaptações	Preserva estrutura inteira
Eficiência Computacional	Moderada	Alta (problemas pequenos/médios)	Alta (problemas grandes)

A transformação da variável inversa, como vimos antes, atua no denominador da função objetivo e gera estabilidade numérica moderada para problemas de PLI. A de Charnes-Cooper, por sua vez, reformula completamente o problema, escalando todas as variáveis pelo fator contínuo adicionado, proporcionando estabilidade numérica mais elevada e compatibilidade direta com PLI, apesar de aumentar o número de variáveis. Talvez por isso seja mais adequada para problemas pequenos e médios (CHARNES; COOPER, 1962). O Dinkelbach, que veremos a seguir, é iterativo, mas não adiciona variáveis extras por iteração, como o nome parece sugerir[12], porquanto preserva a estrutura inteira do problema, sendo eficiente especialmente para problemas de grande porte, apesar de sua estabilidade depender da convergência do algoritmo (DINKELBACH, 1967), como explicaremos no próximo capítulo. Essa comparação, ainda que simples, contribui, do ponto de vista prático, para a escolha da técnica mais adequada, de acordo com o tamanho do problema, da estrutura das variáveis e da necessidade de estabilidade numérica ou

[12] O método de Dinkelbach é iterativo porque resolve uma sequência de problemas auxiliares, ajustando um parâmetro a cada iteração até atingir a convergência. No entanto, em cada iteração, o problema resolvido mantém exatamente as mesmas variáveis e restrições do problema original, apenas modificando a função objetivo. Assim, o método não adiciona variáveis extras por iteração, o que contribui para sua eficiência e simplicidade estrutural, especialmente em problemas de grande porte. Por outro lado, sua estabilidade depende da convergência do algoritmo, que pode variar conforme o problema (DINKELBACH, 1967).

não.

7.5.6 O Algoritmo Iterativo de Dinkelbach

Por fim, chegamos ao Dinkelbach — uma ferramenta frequentemente descrita na literatura como "poderosa" para a resolução de problemas de programação fracionária. O motivo? Ele transforma a complexidade da razão em uma sequência de problemas paramétricos mais simples, todos linearizados, todos tratáveis por solvers modernos. Sua principal virtude, segundo o autor do método, está na capacidade de convergir rapidamente para a solução ótima do problema original, mesmo em casos de grande porte — como algumas das instâncias que enfrentamos neste trabalho.

Partindo do nosso problema fracionário:

$$\max Z = \frac{N(x,y)}{D(x,y)},$$

onde $N(x,y)$ representa a quantidade total de unidades coletadas — e $D(x,y)$ o total de corredores visitados, a ideia central do método de Dinkelbach é transformar o problema fracionário em uma sequência de problemas paramétricos, mais simples de resolver (DINKELBACH, 1967; YOU; GROSSMANN, 2009). Em vez de maximizar a razão diretamente, o método propõe resolver iterativamente um problema auxiliar $P(\lambda)$, onde λ é um parâmetro que representa uma estimativa do valor ótimo da função objetivo fracionária, ajustado a cada passo do algoritmo, até que a função objetivo auxiliar atinja zero (convergência), indicando que a solução encontrada é a solução ótima do problema original (DINKELBACH, 1967).

Formalmente, o problema auxiliar em cada iteração é dado por:

$$P(\lambda): \quad \max \quad f(\lambda) = N(\mathbf{x},\mathbf{y}) - \lambda D(\mathbf{x},\mathbf{y}), \tag{6}$$

e a condição de parada clássica do algoritmo é

$$N(x^*,y^*) - \lambda^* D(x^*,y^*) = 0,$$

o que implica que

$$\lambda^* = \frac{N(x^*,y^*)}{D(x^*,y^*)},$$

isto é, λ^* é o valor ótimo da função objetivo fracionária original.

Para o nosso problema, a função objetivo paramétrica em cada iteração assume a seguinte forma:

$$\max \quad \sum_{o\in\mathcal{O}} S_o \cdot x_o - \lambda\left(1 + \sum_{a\in\mathcal{A}} y_a\right), \tag{7}$$

sujeita a todas as restrições do modelo, como os limites de unidades na wave, a cobertura de itens, entre outras condições previamente definidas. É importante observar que, em cada iteração, o problema $P(\lambda)$ permanece linear, o que permite sua resolução por solvers comerciais como o CPLEX — utilizado neste estudo. Outra vantagem é que ele não requer a introdução de variáveis adicionais a cada iteração, o que preserva a estrutura inteira do problema original e que o torna particularmente eficiente em instâncias de grande porte. No entanto, vale destacar que a estabilidade numérica das soluções obtidas depende diretamente da boa convergência[13] do algoritmo. Quando o método converge adequadamente para o valor ótimo, os resultados são estáveis e confiáveis. Caso contrário, instabilidades numéricas podem surgir e comprometer a precisão das soluções, dificultando sua interpretação ou aplicação prática (DINKELBACH, 1967; DUARTE, 2019).

Para facilitar a compreensão, apresentamos abaixo o pseudocódigo clássico do Dinkelbach, adaptado para a resolução do nosso problema. As adaptações consistem principalmente na especificação das funções $N(x, y)$ e $D(x, y)$, e na inclusão das restrições do problema que devem ser consideradas em cada iteração do solver de PLI. O núcleo do algoritmo permanece o mesmo: **resolver iterativamente um problema linear paramétrico e ajustar o parâmetro λ até a convergência.**

Algoritmo 3 Algoritmo de Dinkelbach para o WOP

Require: Conjunto de pedidos $\mathcal{O}$, corredores $\mathcal{A}$, unidades por pedido S_o, unidades de item por pedido U_{oi}, disponibilidade por corredor AV_{ai}, limites LB, UB, tolerância ε, máximo de iterações MaxIter

Ensure: Valor ótimo λ^*, solução ótima (x^*, y^*)

Inicialize $\lambda_0 \leftarrow 0$, $k \leftarrow 0$

repeat

    $k \leftarrow k + 1$

    Resolva o problema paramétrico:

$$\max \sum_{o\in\mathcal{O}} S_o x_o - \lambda_k \left(1 + \sum_{a\in\mathcal{A}} y_a\right)$$

    sujeito a:

- $\sum_{o\in\mathcal{O}} S_o x_o \geq LB$
- $\sum_{o\in\mathcal{O}} S_o x_o \leq UB$
- $\sum_{o\in\mathcal{O}} U_{oi} x_o \leq \sum_{a\in\mathcal{A}} AV_{ai} y_a,\ \forall i \in \mathcal{I}$
- $x_o \in \{0, 1\}, \forall o \in \mathcal{O}$
- $y_a \in \{0, 1\}, \forall a \in \mathcal{A}$

    Obtenha a solução ótima (x^*, y^*) e valor de $f(\lambda_k)$

    Calcule:

$$N^* = \sum_{o\in\mathcal{O}} S_o x_o^*, \quad D^* = 1 + \sum_{a\in\mathcal{A}} y_a^*$$

    if $|f(\lambda_k)| < \varepsilon$ ou $k \geq$ MaxIter **then**

        Pare

    end if

    Atualize: $\lambda_{k+1} \leftarrow \frac{N^*}{D^*}$

until convergência

return λ_k e (x^*, y^*)

Vamos ao passo a passo:

[13] O método de Dinkelbach é provadamente convergente e, na prática, bastante rápido:

"$q_{k+1} > q_k \ \ldots\ \lim_{k\to\infty} q_k = q^*$"

Isso significa que o parâmetro λ (ou q) avança monotonicamente em direção ao valor ideal, com comportamento estável e superlinear na maioria das aplicações práticas (DINKELBACH, 1967).

1. **Inicialização:** começamos com um valor inicial para o parâmetro λ, como $\lambda_0 = 0$ ou uma estimativa baseada nos valores máximos esperados do numerador e do denominador. Esse valor funciona como um palpite inicial da razão ótima a ser alcançada.

2. **Construção do Problema Paramétrico:** a cada iteração, o algoritmo constrói e resolve um problema auxiliar, substituindo o objetivo fracionário por uma função linear dependente de λ_k:
$$f(\lambda_k) = N(x_k) - \lambda_k D(x_k),$$
onde $N(x)$ e $D(x)$ representam, respectivamente, o numerador e o denominador da função objetivo original. As variáveis x_k indicam a solução corrente — por exemplo, a combinação de pedidos e corredores no nosso contexto.

3. **Resolução do Problema Auxiliar:** resolvemos o problema paramétrico $P(\lambda_k)$, ou seja, encontramos o vetor x_k que maximiza a expressão acima, respeitando todas as restrições do modelo original.

4. **Atualização do Parâmetro:** a partir da solução ótima obtida, calculamos um novo valor para λ:
$$\lambda_{k+1} = \frac{N(x_k)}{D(x_k)}.$$
Esse valor representa uma nova estimativa da razão ideal, baseada na melhor solução encontrada até então.

5. **Critério de Parada:** o processo se repete até que a função auxiliar $f(\lambda_k)$ esteja suficientemente próxima de zero — isto é,
$$|f(\lambda_k)| = |N(x_k) - \lambda_k D(x_k)| < \varepsilon,$$
para uma tolerância pequena $\varepsilon > 0$, previamente definida. Esse valor controla a precisão desejada para a solução final. Quando essa condição é satisfeita, entendemos que a razão atingiu o valor ótimo buscado:
$$\lambda_k \approx \frac{N(x_k)}{D(x_k)}.$$
O método, então, é interrompido, e x_k é considerado uma solução ótima (ou suficientemente boa) para o problema fracionário original.

Isso demonstra que o algoritmo de Dinkelbach segue uma lógica simples, como defende a literatura, mas para entendê-la melhor, como cada passo se traduz na prática — e especialmente por que essa abordagem funciona tão bem —, vale revisitar o nosso já conhecido exemplo do armazém, refazendo os passos do algoritmo nele, dando forma concreta aos conceitos abstratos apresentados.

Imagine, mais uma vez, que você precisa coletar mercadorias no grande armazém do Mercado Livre e está com pressa, cansado de andar, afinal, esta já é a terceira vez que te pedimos isso. Seu objetivo, portanto, é ser mais eficiente ainda, ou seja, pegar a maior quantidade possível de itens, passando pelo menor número de corredores possível.

Apesar disso, pode ser que na primeira tentativa, você simplesmente tente pegar o máximo de itens possível, ignorando completamente a relação entre esforço e resultado. Mas daí você lembra do Dinkelbach e decide aplicá-lo para otimizar sua coleta.

Você começa com um palpite inicial: todo item coletado vale tanto quanto o custo de atravessar um corredor. Então, em vez de tentar maximizar a razão diretamente, você resolve um problema auxiliar em que o objetivo é maximizar a soma das unidades coletadas menos o custo proporcional aos corredores visitados — neste caso, considerando esse custo igual a 1. Isso significa que você calcula o total de unidades coletadas nos pedidos e subtrai o valor correspondente a atravessar os corredores utilizados[14].

Após resolver esse problema, você obtém uma solução inicial com um conjunto de pedidos selecionados e os corredores correspondentes visitados. Com base nisso, você calcula o valor do numerador (quantidade total de unidades coletadas) e o valor do denominador (número de corredores visitados mais um). Com esses dois valores em mãos, você atualiza o parâmetro lambda (λ), que passa a ser a razão entre o total de itens e o total de corredores (ajustado).

Digamos que você coletou 10 itens e percorreu 2 corredores, então o novo valor de lambda será 10 dividido por 3, ou aproximadamente 3,33. Isso significa que, na próxima rodada, cada unidade coletada valerá 3,33 vezes o custo de passar por um corredor — uma forma de ajustar dinamicamente o peso relativo de cada parte da razão.

Com esse novo valor, você repete o processo: resolve novamente o problema auxiliar, agora usando o lambda atualizado como multiplicador do número de corredores. O objetivo continua sendo maximizar o total ajustado da função, levando em conta essa nova ponderação.

Esse processo se repete iterativamente: resolve-se o problema com o lambda atual, calcula-se uma nova razão com a solução encontrada e atualiza-se lambda. O ciclo continua até que a diferença entre o numerador e o denominador ponderado fique suficientemente próxima de zero — ou seja, até que a função auxiliar pare de melhorar significativamente. Quando isso ocorre, significa que a razão ótima foi atingida, e a solução encontrada é considerada ótima para o problema fracionário original (DINKELBACH, 1967).

Ao final desse processo, você — cansado, mas satisfeito — percebe que percorreu o mínimo necessário do armazém, colheu a melhor quantidade possível de pedidos e fez tudo isso com uma eficiência que nem mesmo os algoritmos de roteamento do Mercado Livre conseguiriam bater. E o melhor: sem precisar testar todas as combinações possíveis ou ajustar manualmente pesos e penalidades.

Essa jornada iterativa, que começou com um palpite e evoluiu com ajustes progressivos da razão entre esforço e recompensa, é exatamente o que o algoritmo de Dinkelbach propõe — e o que o torna uma das abordagens mais interessantes e poderosas para resolver o nosso problema. Ufa! Agora que compreendemos os principais métodos para lidar com funções objetivo fracionárias, podemos apresentar a escolha metodológica que orientou nossa modelagem final para o Desafio SBPO 2025.

Mais do que uma decisão técnica, a seleção dos métodos adotados levou em conta fatores práticos como estabilidade numérica, custo computacional, escalabilidade e compatibilidade com os solvers disponíveis (em especial, o IBM CPLEX). Além disso, exploramos como a combinação estratégica entre esses métodos e o uso de aceleração via GPU (CUDA) pode oferecer vantagens concretas em instâncias de grande porte do Wave Order Picking (WOP). Vamos, portanto, ao que realmente interessa: como isso tudo se traduziu, na prática, na escolha e na implementação que fizemos.

7.5.7 Escolha e Implementação no Contexto do WOP

Compreendidas as diferentes técnicas de linearização, esta seção consolidará a discussão apresentando a escolha final e a justificativa para as abordagens adotadas em nosso trabalho. Discutiremos como a combinação estratégica de métodos específicos, em sinergia com a integração do *solver* IBM CPLEX e a aceleração por NVIDIA CUDA, foi fundamental para alcançar soluções ótimas dentro dos rigorosos limites de tempo do Desafio SBPO 2025.

[14] Com o ajuste de robustez somando 1 ao total de corredores visitados. Vide nota 3.

8 Metodologia da Solução Proposta

Esta seção detalha a abordagem metodológica adotada para resolver o Problema de Wave Order Picking (WOP) proposto no Desafio SBPO 2025 do Mercado Livre. O núcleo da metodologia é um modelo de Programação Inteira Mista (MILP) projetado para capturar as especificidades do problema e encontrar soluções ótimas.

8.1 Arquitetura Geral da Solução

A implementação da solução segue uma arquitetura modular, com scripts Python dedicados a diferentes etapas do processo, desde a leitura dos dados até a validação da solução.

8.2 Pré-processamento de Dados e do Modelo

O pré-processamento é uma etapa crucial para a eficiência da solução, dividindo-se em pré-processamento de dados (organização para acesso rápido) e pré-processamento do modelo (redução da complexidade do problema).

8.2.1 Pré-processamento de Dados

O primeiro passo é a leitura e interpretação dos dados da instância, seguida pela organização em estruturas eficientes.

8.2.2 Técnicas de Pré-processamento do Modelo (Redução do Problema)

Além da organização dos dados, técnicas de pré-processamento do modelo são aplicadas para reduzir o tamanho e a complexidade do problema antes de submetê-lo ao solver.

8.3 Tratamento de Restrições (Rígidas e Suaves)

A implementação permite a escolha entre restrições rígidas, que devem ser obrigatoriamente satisfeitas, e restrições suaves, que permitem violações mediante penalidades na função objetivo.

8.4 Integração e Aceleração Computacional (CPLEX e CUDA)

A viabilidade da abordagem exata para o WOP foi drasticamente aprimorada pela integração com o solver CPLEX e a aceleração computacional via NVIDIA CUDA.

8.5 Processo de Resolução (Branch-and-Cut)

Uma vez que o modelo de PLI é construído e linearizado, o solver CPLEX emprega algoritmos avançados para encontrar a solução ótima.

8.6 Extração e Validação da Solução

Após a resolução pelo solver, a solução obtida é extraída e validada para garantir sua viabilidade e calcular o valor da função objetivo oficial.

9 Experimentos e Resultados

Nesta seção, apresentaremos os resultados computacionais obtidos pela nossa abordagem de Programação Linear Inteira (PLI) para o Problema, utilizando as instâncias do Desafio SBPO 2025 do Mercado Livre. O objetivo é avaliar a eficácia da solução exata, o impacto das diferentes técnicas de linearização, a influência das configurações de restrições (rígidas vs. suaves), o papel crucial das técnicas de pré-processamento e da aceleração com CUDA, e comparar o desempenho com os resultados de referência do desafio ("BOV Oficial- Melhor Valor Objetivo Oficial).

9.1 Configuração Experimental

Detalhes sobre o ambiente de hardware e software, bem como os parâmetros utilizados nos experimentos.

9.2 Análise dos Resultados

Análise dos resultados obtidos com as configurações "oficiais"em comparação com o Melhor Valor Objetivo (BOV) oficial do desafio.

9.3 Análise dos Benchmarks de Configuração

Avaliação do desempenho das diferentes configurações testadas, incluindo o impacto do Big-M e do uso de GPU.

9.4 O Papel Crucial do Pré-processamento e da Aceleração CUDA

Reafirmação e detalhamento do impacto fundamental das técnicas de pré-processamento e da aceleração CUDA na viabilidade da abordagem exata.

10 Discussão

A decisão de empregar um método exato (PLI) para um problema NP-Difícil como o WOP, em detrimento de abordagens heurísticas, é justificada pelos resultados obtidos e pelas capacidades da implementação.

10.1 Vantagens da Abordagem Exata

Análise das principais vantagens de utilizar uma abordagem exata para o problema do WOP.

10.2 Viabilidade Através de Pré-processamento e CUDA

Elaboração sobre como as técnicas de pré-processamento e a aceleração CUDA tornaram a abordagem exata computacionalmente viável.

10.3 Comparação Implícita com Heurísticas

Discussão sobre a competitividade da abordagem exata em relação às heurísticas, mesmo sem uma comparação direta explícita.

10.4 Linearização e Flexibilidade do Modelo

Análise da importância da escolha dos métodos de linearização e da flexibilidade do modelo (restrições rígidas/suaves).

10.5 Limitações da Abordagem

Identificação das limitações inerentes à abordagem exata, mesmo com as otimizações implementadas.

11 Conclusão

Este trabalho apresentou uma solução exata competitiva para o Problema de Wave Order Picking do Desafio SBPO 2025, utilizando Programação Linear Inteira com o solver CPLEX e CUDA.

11.1 Sumário das Contribuições

Resumo das principais contribuições do estudo.

Referências

CK-12 FOUNDATION. **O que é uma função objetivo no contexto da programação linear?** [S.l.: s.n.], n.d. Disponível em: <https://www.ck12.org/flexi/pt-br/algebra/programacao-linear/o-que-e-uma-funcao-objetivo-no-contexto-da-programacao-linear/>. Acesso em: 28 mai. 2024.

ACHTERBERG, Tobias. **Constraint Integer Programming**. 2007. PhD thesis – Technische Universit"at Berlin. Disponível em: <https://opus4.kobv.de/opus4-zib/frontdoor/index/index/docId/1112>.

ACHTERBERG, Tobias et al. Presolve Reductions in Mixed Integer Programming. **INFORMS Journal on Computing**, INFORMS, v. 32, n. 2, p. 473–506, 2020. DOI: 10.1287/ijoc.2019.0895.

ÁGUIA SISTEMAS. **Erros de Picking no Armazém: Não com a Linha de Automação da Águia Sistemas**. [S.l.: s.n.], 2024. Disponível em: <https://aguiasistemas.com.br/erros-de-picking-no-armazem-nao-com-a-linha-de-automacao-da-aguia-sistemas/>.

ALMEIDA, Mariano; REBELATTO, Daniel. **Princípios básicos para uma proposta de ensino sobre análise por envoltória de dados**. [S.l.: s.n.], 2019. Anais do XXXIV Congresso Brasileiro de Ensino de Engenharia. Disponível em: <https://www.abenge.org.br/cobenge/legado/arquivos/13/artigos/14_285_716.pdf>.

APPLEGATE, David L. et al. **Primal-Dual Hybrid Gradient Methods for Large-Scale Optimization**. [S.l.: s.n.], 2021. https://arxiv.org/abs/2103.00000. Acesso em: 27 maio 2025.

BALAS, Egon. Disjunctive Programming: Properties of the Convex Hull of Feasible Points. **Discrete Applied Mathematics**, v. 89, n. 1-3, p. 3–44, 1998. DOI: 10.1016/S0166-218X(97)00102-5.

BEN-TAL, Aharon; EL GHAOUI, Laurent; NEMIROVSKI, Arkadi. **Robust Optimization**. 1st. [S.l.]: Princeton University Press, 2009. ISBN 978-0691135296.

BERTSIMAS, Dimitris; SIM, Melvyn. **Robust Optimization**. 1st. Philadelphia: SIAM, 2011. ISBN 978-0898718823.

________. The Price of Robustness. **Operations Research**, v. 52, n. 1, p. 35–53, 2004. DOI: 10.1287/opre.1030.0065.

BIRGIN, Ernesto G.; MARTÍNEZ, José Mario. **Practical Augmented Lagrangian Methods for Constrained Optimization**. [S.l.]: SIAM, 2014.

BOUERI, Rodrigo; ROCHA, Eduardo; RODOPOULOS, Fernando. Análise Envoltória de Dados (DEA) nas produções acadêmicas. In: XXVI Congresso Brasileiro de Custos. [S.l.: s.n.], 2019. P. 1–10. Disponível em: <https://racef.fundace.org.br/index.php/racef/article/download/332/80>.

BOZ, E.; ARAS, N. The order batching problem: A state-of-the-art review. **Sigma Journal of Engineering and Natural Sciences**, v. 40, n. 2, p. 402–420, 2022. DOI: 10.14744/sigma.2022.00018. Disponível em: <https://dergipark.org.tr/en/download/article-file/2470183>.

BOZ, Emrah; ARAS, Necati. The order batching problem: A state-of-the-art review. **Sigma Journal of Engineering and Natural Sciences**, v. 40, n. 2, p. 402–420, 2022. DOI: 10.14744/sigma.2022.00018. Disponível em: <https://dergipark.org.tr/en/download/article-file/2470183>.

BURACHIK, Regina S.; KAYA, C. Yalcin. An augmented Lagrangian method for conic optimization problems. **Journal of Optimization Theory and Applications**, Springer, v. 155, n. 1, p. 267–282, 2012.

ÇAĞRC, Merve. **MILP FORMULATIONS FOR THE ORDER BATCHING PROBLEM IN LOW-LEVEL PICKER-TO-PART WAREHOUSE SYSTEMS**. Abr. 2014. Master of Science Thesis – Middle East Technical University. Supervisor: Assoc. Prof. Dr. Temel "Oncan. Disponível em: <https://acikbilim.yok.gov.tr/bitstream/handle/20.500.12812/86973/yokAcikBilim_10035380.pdf?sequence=-1&isAllowed=y>.

CALAFATE. **Automatização do Processo de Picking por Ondas [wave picking]**. 2020. Disponível em: <https://pt.linkedin.com/pulse/automatizacao-do-processo-de-picking-por-ondas-wave-com-calafate>. Acesso em: 27 mai. 2025.

CHARNES, Abraham; COOPER, William W. Programming with Linear Fractional Functionals. **Naval Research Logistics Quarterly**, Wiley Online Library, v. 9, n. 3-4, p. 181–186, 1962. DOI: 10.1002/nav.3800090303. Disponível em: <https://onlinelibrary.wiley.com/doi/10.1002/nav.3800090303>.

CHARNES, Abraham; COOPER, William W.; RHODES, Edward. Measuring the Efficiency of Decision Making Units. **European Journal of Operational Research**, v. 2, n. 6, p. 429–444, 1978.

COELLI, Tim; RAO, D.S. Prasada; BATTESE, George E. **An Introduction to Efficiency and Productivity Analysis**. [S.l.]: Kluwer Academic Publishers, 1998.

COIN-OR. **COIN-OR Branch and Cut (CBC) Solver**. [S.l.: s.n.], 2020. https://github.com/coin-or/Cbc.

COLUMBIA UNIVERSITY. **4.10 – The Big M Method**. [S.l.: s.n.], n.d. Acesso em: 28 maio 2025. Disponível em: <http://www.columbia.edu/~cs2035/courses/ieor3608.F05/david-bigM.pdf>.

DCC/UFMG. **Problemas NP-Completo e Algoritmos Aproximados**. [S.l.: s.n.]. https://www2.dcc.ufmg.br/livros/algoritmos/cap9/slides/pascal/completo4/cap9.pdf. Acesso em: 27 maio 2025.

DESCONHECIDO, Autor. **Metaheuristics for the Order Batching Problem in Manual Order Picking Systems**. [S.l.: s.n.], 2018. https://www.econstor.eu/bitstream/10419/103688/1/2508.pdf. Acesso em: 27 maio 2025.

DINKELBACH, W. On nonlinear fractional programming. **Management Science**, v. 13, n. 7, p. 492–498, 1967. DOI: 10.1287/mnsc.13.7.492.

DUARTE, Luís Henrique Rodrigues. **Aplicação de métodos biobjetivo à otimização linear fracionária**. 2019. Dissertação de Mestrado em Matemática – Universidade de Coimbra, Coimbra, Portugal. Disponível em: https://hdl.handle.net/10316/87827 e https://estudogeral.uc.pt/bitstream/10316/87827/1/thesis.pdf. Disponível em: <https://hdl.handle.net/10316/87827>.

E-SHIP WMS. **O que é Picking por Onda e Pré-Picking?** [S.l.: s.n.], 2024. https://eship.com.br/o-que-e-picking-por-onda-e-pre-picking/. Acesso em: 27 maio 2025.

EUROCC NCC BELGIUM. **Order batching problems in warehouse are NP-hard**. [S.l.: s.n.]. https://www.enccb.be/uslogisticsnphardness. Acesso em: 27 maio 2025.

FACOM-UFMS. **Programação Linear - Material de Curso**. [S.l.: s.n.], 2009. http://www.facom.ufms.br/~ricardo/Courses/OR-2009/Materials/plinear.pdf.

FERREIRA, Jonas. **Modelos de análise envoltória de dados e suas aplicações**. 2015. Tese (Doutorado) – Universidade de São Paulo. Disponível em: <https://www.teses.usp.br/teses/disponiveis/96/96132/tde-23102015-111731/publico/JonasFerreira_Corrigida.pdf>.

FISHER, Marshall L. The Lagrangian relaxation method for solving integer programming problems. **Management Science**, INFORMS, v. 50, n. 12, p. 1861–1871, 2004.

GADEMANN, Niels; VELDE, Sander van de. Order batching to minimize total travel time in a parallel-aisle warehouse. **European Journal of Operational Research**, v. 161, n. 3, p. 668–693, 2005. DOI: 10.1016/j.ejor.2003.08.011. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0377221703004363>.

GEOFFRION, Arthur M. Lagrangean relaxation for integer programming. **Mathematical Programming Study**, Springer, v. 2, p. 82–114, 1974.

GOSHIP. **G+Ship: logística para e-commerce em Porto Alegre**. [S.l.: s.n.], 2024. Disponível em: <https://goship.com.br/rio-grande-do-sul/porto-alegre/>.

GUIGNARD, Michel. Lagrangean relaxation. **TOP**, Springer, v. 11, n. 2, p. 151–200, 2003.

GUROBI OPTIMIZATION, LLC. **Gurobi Optimizer Reference Manual**. [S.l.: s.n.], 2023. https://www.gurobi.com.

HENN, Sebastian; KOCH, Sören; DOERNER, Karl F. et al. Metaheuristics for the Order Batching Problem in Manual Order Picking Systems. **BuR - Business Research**, v. 3, n. 1, p. 82–105, 2010. DOI: 10.1007/BF03342717. Disponível em: <https://www.econstor.eu/bitstream/10419/103688/1/2508.pdf>.

HENN, Sebastian; KOCH, Stefan; WÄSCHER, Gerhard. Order batching in order picking warehouses: A survey of solution approaches. **International Journal of Production Research**, Taylor & Francis, v. 50, n. 3, p. 779–802, 2012. DOI: 10.1080/00207543.2010.538744.

IBM. **IBM ILOG CPLEX Optimization Studio**. [S.l.: s.n.], 2017. https://www.ibm.com/products/ilog-cplex-optimization-studio.

IBM CORPORATION. **IBM ILOG CPLEX Optimization Studio**. [S.l.: s.n.], 2024. https://www.ibm.com/products/ilog-cplex-optimization-studio. Acesso em: 27 maio 2025.

________. **O que é modelagem de otimização?** [S.l.: s.n.], 2023. Disponível em: <https://www.ibm.com/br-pt/think/topics/optimization-model>. Acesso em: 28 mai. 2024.

INVIA ROBOTICS. **Guide To Wave Warehouse Order Picking**. [S.l.: s.n.], 2024. Disponível em: <https://inviarobotics.com/blog/guide-to-wave-warehouse-order-picking/>.

KOLMAN, Bernard. **Introdução à Programação Linear**. [S.l.]: Editora XYZ, 1995.

LUANDA. **Método Grande M em Programação Linear**. [S.l.: s.n.], 2022. Trabalho acadêmico disponível em repositório institucional.

MANHATTAN ASSOCIATES. **Warehouse Management System (WMS)**. [S.l.: s.n.], 2024. https://www.manh.com/solutions/supply-chain-management-software/warehouse-management/wes-inside-wms. Acesso em: 27 maio 2025.

MARLER, R. T.; ARORA, J. S. Survey of multi-objective optimization methods for engineering. **Structural and Multidisciplinary Optimization**, v. 26, n. 6, p. 369–395, 2004. DOI: 10.1007/s00158-003-0368-6.

MARQUES JÚNIOR, Francisco Daladier. **Modelagem e Análise de Eficiência em Redes Virtuais Usando DEA e Modelos Fractais**. 2021. Tese (Doutorado) – Universidade Federal de Pernambuco. Disponível em: <https://repositorio.ufpe.br/bitstream/123456789/35199/1/TESE%20Francisco%20Daladier%20Marques%20J%C3%BAnior.pdf>.

MAXWELL. **Análise Envoltória de Dados**. [S.l.: s.n.], 2014. https://www.maxwell.vrac.puc-rio.br/16033/16033_3.PDF. Disponível em: https://www.maxwell.vrac.puc-rio.br/16033/16033_3.PDF, acesso em 30 maio 2025.

MCCORMICK, Garth P. Computability of global solutions to factorable nonconvex programs. **Mathematical Programming**, v. 10, p. 147–175, 1976.

MECALUX. **Quando aplicar o picking por ondas ou wave picking**. [S.l.: s.n.], 2020. https://www.mecalux.pt/blog/picking-por-onda. Acesso em: 27 maio 2025.

MEDEIROS, José Maurício Fernandes. **Abordagens heurísticas para o problema de escalonamento de lotes em máquinas paralelas com penalidades por atraso**. 2023. Tese (Doutorado) – Universidade Federal da Paraíba. Disponível em: <https://repositorio.ufpb.br/jspui/handle/123456789/31091>.

MERCADO LIVRE. **Repositório do Desafio no Github**. [S.l.: s.n.], 2025. https://github.com/mercadolibre/challenge-sbpo-2025. Acesso em: 27 abril de 2025.

MIETTINEN, Kaisa. **Nonlinear Multiobjective Optimization**. [S.l.]: Springer, 1999. ISBN 978-0792374207.

MIT. **Learning-Augmented Optimization**. [S.l.: s.n.], 2020. https://mit.edu/learning-augmented-optimization. Acesso em: 27 maio 2025.

MIT OPENCOURSEWARE. **IP Reference guide for integer programming formulations**. [S.l.], 2013. Acesso em: 28 maio 2025. Disponível em: <https://ocw.mit.edu/courses/15-053-optimization-methods-in-management-science-spring-2013/86d472ffa3f1c341c586cb26ba1093c1_MIT15_053S13_iprefguide.pdf>.

NEMHAUSER, George L; WOLSEY, Laurence A. **Integer and Combinatorial Optimization**. New York: Wiley-Interscience, 1988. ISBN 0-471-82819-X.

NETSUITE. **What Is Wave Picking? How It Works, Methods & Tips**. [S.l.: s.n.], 2025. Disponível em: <https://www.netsuite.com/portal/resource/articles/inventory-management/wave-picking.shtml>.

NOCEDAL, Jorge; WRIGHT, Stephen J. **Numerical Optimization**. 2nd. [S.l.]: Springer, 2006. ISBN 978-0387303031.

NVIDIA CORPORATION. **NVIDIA cuOpt - GPU-Accelerated Decision Optimization Solver**. [S.l.: s.n.], 2024. https://www.nvidia.com/en-us/ai-data-science/products/cuopt/. Acesso em: 27 maio 2025.

ÖNCAN, T. Mathematical models for order batching in warehouses. **International Journal of Production Research**, v. 53, n. 11, p. 3388–3402, 2015. DOI: 10.1080/00207543.2014.993958. Disponível em: <https://doi.org/10.1080/00207543.2014.993958>.

ÖNCAN, Temel. Problem in Low-Level Picker-to-Part Warehouse Systems. In: PROCEEDINGS of the International MultiConference of Engineers and Computer Scientists (IMECS). [S.l.: s.n.], 2013. P. 19–24. Disponível em: <https://www.iaeng.org/publication/IMECS2013/IMECS2013_pp19-24.pdf>.

PANSART, Jean-François; DAUZÈRE-PÉRÈS, Stéphane; GOURGAND, Michel. A sparse MILP formulation for the order batching problem in warehouses. **European Journal of Operational Research**, v. 271, n. 3, p. 1076–1089, 2018. DOI: 10.1016/j.ejor.2018.06.049. Disponível em: <https://doi.org/10.1016/j.ejor.2018.06.049>.

PEDROSO, João Pedro. **Método do Big-M e suas aplicações em Programação Linear**. [S.l.], 2024. Material didático disponível em aula de Métodos de Apoio à Decisão.

PESQUISA OPERACIONAL, Sociedade Brasileira de. **Primeiro Desafio Mercado Livre de Otimização**. [S.l.: s.n.], 2025. Reprodução autorizada do artigo de Cid Carvalho de Souza. Disponível em: <https://sbpo2025.galoa.com.br/sbpo-2025/page/5786-primeiro-desafio-mercado-livre-de-otimizacao?lang=pt-br>. Acesso em: 27 abr. 2025.

PLATINUM LOG. **Conheça as 7 etapas da armazenagem**. [S.l.: s.n.], 2023. https://www.tpl.com.br/blog/conheca-as-7-etapas-da-armazenagem. Acesso em: 27 maio 2025.

POCINHO, Daniel. **Formulações de Linearização por Partes para Programação Linear Inteira Mista: Aplicação à Alocação Ótima de Gás de Injeção**. 2020. Tese (Doutorado) – Universidade Federal de Santa Catarina. Disponível em: <https://repositorio.ufsc.br/handle/123456789/2020>.

POCINHO, Guilherme Fialho Costa. **Análise e melhoria do processo de order-picking num sistema produtivo: caso de estudo**. 2013. Tese (Doutorado) – Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa. Disponível em: <https://run.unl.pt/bitstream/10362/11038/1/Pocinho_2013.pdf>.

PUC-RIO, DEPARTAMENTO DE ENGENHARIA INDUSTRIAL. **Métodos de Otimização**. [S.l.: s.n.], 2011. https://www2.dbd.puc-rio.br/pergamum/tesesabertas/0310953_05_cap_03.pdf. Acesso em: 27 maio 2025.

SAP SE. **SAP Extended Warehouse Management (SAP EWM)**. [S.l.: s.n.], 2023. https://help.sap.com/doc/saphelp_ewm700_ehp02/7.0.2/en-US/f1/704c06fae84e1085867e3452f5a111/content.htm. Acesso em: 27 maio 2025.

SCHAIBLE, Siegfried. Fractional Programming: Applications and Algorithms. **European Journal of Operational Research**, Elsevier, v. 7, n. 2, p. 111–120, 1981. DOI: 10.1016/0377-2217(81)90272-0. Disponível em: <https://www.sciencedirect.com/science/article/pii/0377221781902721>.

SCHAIBLE, Siegfried; IBARAKI, Toshihide. Fractional Programming. **European Journal of Operational Research**, Elsevier, v. 12, n. 4, p. 325–338, 1983. DOI: 10.1016/0377-2217(83)90153-9. Disponível em: <https://www.sciencedirect.com/science/article/pii/0377221783901534>.

SENIOR SISTEMAS. **O que é otimização logística e qual sua importância?** [S.l.: s.n.], 2023. Disponível em: <https://www.senior.com.br/blog/otimizacao-logistica>. Acesso em: 28 mai. 2024.

SHERALI, Hanif D.; ADAMS, Warren P. **A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems**. [S.l.]: Springer, 1999.

SHERALI, Hanif D.; FRATICELLI, Bruno. A New Piecewise-Linear Approach for Solving Nonconvex Nonlinear Programming Problems. **Operations Research**, v. 49, n. 2, p. 273–288, 2001. DOI: 10.1287/opre.49.2.273.11283.

SHIAU, Jiun-Yan; HUANG, Jie-An. Wave Planning for Cart Picking in a Randomized Storage Warehouse. **Applied Sciences**, MDPI, Basel, Switzerland, v. 10, n. 8050, 2020. DOI: 10.3390/app10228050.

SOUZA, Cid Carvalho de. **Desafio Mercado Livre de Otimização**. [S.l.: s.n.], 2025. Publicado em Medium. Disponível em: <https://medium.com/mercadolibre-tech/desafio-mercado-livre-de-otimiza%C3%A7%C3%A3o-6f7b1b4d3c5e>. Acesso em: 27 abr. 2025.

SOUZA, Robert de; JESUS GOMES FURTADO, Crispiniano de; HORTA, João Carlos Lopes. Resolução gráfica de um problema de programação linear utilizando a folha gráfica 3D do GeoGebra. **Revista do Instituto GeoGebra de São Paulo**, v. 7, n. 2, p. 45–64, 2018. Disponível em: https://dialnet.unirioja.es/descarga/articulo/6767155.pdf. Disponível em: <https://dialnet.unirioja.es/descarga/articulo/6767155.pdf>.

TAVARES, Ricardo Gonçalves. **Meta-heurísticas para o sequenciamento de famílias de tarefas em máquinas paralelas uniformes de processamento em lote**. 2020. Diss. (Mestrado) – Universidade Federal de Viçosa. Disponível em: <https://locus.ufv.br/handle/123456789/31091>.

TAWARMALANI, Mohit; SAHINIDIS, Nikolaos V. Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software, and Applications. **Springer**, 2002.

VTEX. **Como a satisfação do cliente na logística pode influenciar na recompra**. [S.l.: s.n.], 2020. Disponível em: <https://vtex.com/pt-br/blog/estrategia/satisfacao-do-cliente-na-logistica/>.

WIKIPÉDIA. **NP-difícil**. [S.l.: s.n.], 2010. https://pt.wikipedia.org/wiki/NP-difÃ■cil. Acesso em: 27 maio 2025.

WOLSEY, Laurence A. **Integer Programming**. New York: Wiley-Interscience, 1998. (Wiley-Interscience Series in Discrete Mathematics and Optimization). ISBN 0-471-28366-5.

________.________. 1st. [S.l.]: Wiley-Interscience, 1999. ISBN 978-0471359432.

YOU, Fengqi; GROSSMANN, Ignacio E. Mixed-integer Programming Models and Algorithms for Integrated Capacitated Production Planning and Scheduling in Continuous Manufacturing Industries. **Computers & Chemical Engineering**, Elsevier, v. 33, n. 12, p. 1711–1727, 2009. DOI: 10.1016/j.compchemeng.2009.02.005. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0098135409000551>.

ZHU, Joe. Fractional programming and its applications in data envelopment analysis. **European Journal of Operational Research**, v. 252, n. 3, p. 593–600, 2016. DOI: 10.1016/j.ejor.2015.12.038.

Histórico do Artigo:
Versão 1
Entregue: 31 de maio de 2025