

Spatial Decomposition and Hybrid Optimization for the Online Order Batching Problem under Logistic Constraints

Carlos V. Dantas Araújo¹, Jailon W. B. Oliveira da Silva¹
Pablo L. Braga Soares¹

¹Laboratório de Pesquisa & Desenvolvimento do NEMO
Universidade Federal do Ceará – Campus Russas
Avenida Felipe Santiago – Nº 411 62.900-624, Russas/CE, Brasil

{carlosvinicio, williambrunocc}@alu.ufc.br, pablo.soares@ufc.br

Abstract. *This paper addresses the Online Order Batching Problem (OOBP), a central challenge in distribution center optimization. The proposed hybrid approach combines unsupervised machine learning (density-based spatial clustering) with Multi-Start Simulated Annealing to achieve rapid convergence to high-quality solutions. Validated on 144 benchmark instances from the Albareda-Sambola and Henn datasets, the method achieves solutions with an average gap of 7.9% compared to the state-of-the-art while reducing computational time by 93.2%. The technique solves large-scale instances ($n=250$) in under 60 seconds, demonstrating feasibility for real-time logistics operations.*

Resumo. *Este trabalho aborda o Problema de Loteamento de Pedidos Online, desafio central na otimização de centros de distribuição. A abordagem híbrida proposta combina aprendizado de máquina não supervisionado (clusterização baseada em densidade espacial) com Têmpera Simulada Multi-Start para convergência rápida a soluções de alta qualidade. Validada em 144 instâncias de benchmark dos conjuntos Albareda-Sambola e Henn, o método alcança soluções com gap médio de 7.9% em relação ao estado da arte, enquanto reduz o tempo computacional em 93.2% em comparação ao mesmo. A técnica resolve instâncias grandes ($n=250$) em menos de 60 segundos, demonstrando viabilidade para operações logísticas em tempo real.*

1. Introdução

O Problema de Loteamento de Pedidos Online, do inglês *Online Order Batching Problem* (OOBP) representa um dos desafios centrais na otimização de operações em centros de distribuição, impulsionado pela expansão do e-commerce. O problema consiste em agrupar dinamicamente pedidos de clientes, que chegam continuamente ao sistema, em lotes (batches) para a coleta por operadores (pickers), com o objetivo de minimizar uma função de custo, como o tempo total de conclusão das coletas [Gil-Borrás et al. 2020]. A complexidade do OOBP, classificado como NP-difícil, reside na interdependência das decisões de agrupamento e de roteamento, tornando a busca por soluções ótimas inviável para instâncias de larga escala.

A literatura apresenta diversas abordagens para o Problema de Agrupamento de Pedidos Online (OOBP). Inicialmente, heurísticas simples como *First-Come, First-Served* (FCFS) foram exploradas por sua fácil implementação, embora geralmente resultem em

soluções de qualidade modesta. Em contraste, métodos mais sofisticados consolidaram-se como o estado da arte, a exemplo do *Iterated Local Search* (ILS) proposto por [Henn 2012], que demonstra alta eficácia. Outras técnicas, como o *Estimation of Distribution Algorithm* (EDA) de [Pérez-Rodríguez et al. 2015], também foram aplicadas ao problema, porém com sucesso limitado em superar os *benchmarks* existentes. Recentemente, uma nova fronteira foi explorada com o uso de Aprendizado por Reforço, como no trabalho de [Cals et al. 2020], que aplicaram *Deep Reinforcement Learning* para a decisão de loteamento em tempo real, sinalizando um caminho promissor para a otimização do OOBP.

Apesar da eficácia dos métodos de estado da arte em encontrar soluções de alta qualidade, uma limitação prática emerge: o seu custo computacional. Nossas análises, baseadas em uma reimplementação do ILS, indicam que esse método, apesar de retornar soluções muito boas, pode demandar um tempo de execução superior a 1200 segundos para convergir para as melhores soluções conhecidas em instâncias complexas. Em ambientes de logística em tempo real, onde decisões precisam ser tomadas em minutos, este tempo tende a ser restritivo.

Este trabalho aborda essa lacuna, focando no desenvolvimento de uma abordagem que equilibre o *trade-off* entre a qualidade da solução e o tempo de convergência. Nossa principal contribuição é uma meta-heurística híbrida que utiliza aprendizado de máquina não supervisionado como guia ativo no processo de otimização. Propomos um *framework* que: 1) decompõe o problema usando clusterização sobre *features* de densidade espacial; e 2) utiliza meta-heurística Multi-Start com Têmpera Simulada para otimização rápida.

2. Fundamentação Teórica

Esta seção estabelece os conceitos fundamentais sobre os quais nossa abordagem foi construída, contextualizando sua aplicação ao problema em questão.

2.1. O Problema de Agrupamento de Pedidos Online (OOBP)

O OOBP consiste em agrupar um conjunto de pedidos $\mathcal{O} = \{o_1, o_2, \dots, o_n\}$ que chegam a um armazém ao longo do tempo em um conjunto de lotes $\mathcal{B} = \{b_1, b_2, \dots, b_m\}$, de modo a minimizar uma função objetivo. Cada lote b_j tem uma capacidade máxima W que não pode ser excedida pela soma das demandas dos pedidos nele contidos. A variante que tratamos foca na **minimização** do Tempo de Conclusão Máximo (*makespan*), que é o instante de tempo em que o último lote é coletado e entregue no depósito.

O tempo de serviço para um único lote b_j , $T_{\text{service}}(b_j)$, é definido como a soma de três componentes [Gil-Borrás et al. 2020]:

$$T_{\text{service}}(b_j) = T_{\text{setup}} + T_{\text{routing}}(b_j) + T_{\text{picking}}(b_j) \quad (1)$$

onde:

T_{setup} : Tempo fixo de preparação (configuração do carrinho, análise de pedidos).

T_{routing} : Tempo de deslocamento entre locais de coleta.

T_{picking} : Tempo de coleta/retirada dos itens das prateleiras.

O tempo de roteamento depende da distância percorrida $d(b_j)$ e da velocidade do coletor $v_{routing}$, enquanto o tempo de *picking* depende do número total de itens no lote $N(b_j)$ e da velocidade de coleta $v_{picking}$:

$$T_{routing}(b_j) = \frac{d(b_j)}{v_{routing}} \quad \& \quad T_{picking}(b_j) = \frac{N(b_j)}{v_{picking}} \quad (2)$$

A natureza online do problema introduz a restrição fundamental de que um lote b_j não pode iniciar sua coleta antes do tempo de chegada $t_{arrival}(o_i)$ de todos os pedidos $o_i \in b_j$. Se $t_{start}(b_j)$ é o tempo de início da coleta do lote b_j , a simulação sequencial do tempo de conclusão total C_{max} é dada por:

$$C_{max} = \sum_{j=1}^m \left(\max(t_{completion}(b_{j-1}), \max_{o_i \in b_j} t_{arrival}(o_i)) + T_{service}(b_j) \right) \quad (3)$$

onde $t_{completion}(b_{j-1})$ é o tempo de conclusão do lote anterior.

2.2. Clusterização por K-Means

O K-Means é um algoritmo de aprendizado de máquina não supervisionado que particiona um conjunto de dados em k clusters. O algoritmo visa minimizar a variância intra-cluster, ou a soma dos quadrados das distâncias euclidianas entre cada ponto e o centroide do seu cluster assignado [MacQueen 1967].

Formalmente, dado um conjunto de observações $(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$ onde cada observação é um vetor d -dimensional, o K-Means particiona as n observações em k ($\leq n$) conjuntos $\mathbf{S} = \{S_1, S_2, \dots, S_k\}$ de modo a minimizar a soma dos quadrados intra-cluster:

$$\arg \min_{\mathbf{S}} \sum_{i=1}^k \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2 \quad (4)$$

onde $\boldsymbol{\mu}_i$ é a média dos pontos em S_i .

2.3. Têmpera Simulada

Inspirada no processo metalúrgico de recozimento, a Têmpera Simulada é uma meta-heurística projetada para encontrar o ótimo global de um problema, evitando ficar presa em ótimos locais [Kirkpatrick et al. 1983].

O algoritmo opera através de uma analogia termodinâmica: começa em um estado inicial (solução) com alta “temperatura” T , permitindo movimentos que podem piorar a solução corrente. À medida que a temperatura diminui, o algoritmo torna-se mais seletivo, convergindo para uma solução de alta qualidade. A probabilidade de aceitar uma solução pior é dada por:

$$P(\text{aceitar movimento pior}) = \exp \left(-\frac{\Delta C}{T} \right) \quad (5)$$

onde ΔC é a variação (positiva) do custo.

3. Metodologia

A abordagem proposta para o OOBP neste trabalho é de um sistema de várias fases projetado para lidar com a alta complexidade do problema, priorizando a rápida convergência para soluções de alta qualidade.

3.1. Conjunto de Dados e Cenário Experimental

Para a validação experimental, utilizamos dois conjuntos de instâncias de *benchmark* amplamente reconhecidos na literatura:

Conjunto Henn [Henn 2012]: Este conjunto consiste em 64 instâncias baseadas em um armazém retangular com 10 corredores e 90 locais de armazenamento cada. O depósito está sempre localizado no canto inferior esquerdo. As instâncias apresentam duas características principais:

- **Políticas de armazenamento:**
 - *ABC*: Distribuição onde os itens mais populares (80% da demanda) ocupam 20% dos locais mais acessíveis
 - *Aleatória*: Distribuição uniforme de itens pelos locais de armazenamento
- **Variação de parâmetros:**
 - Pedidos de clientes: 40, 60, 80, 100
 - Capacidade do coletor: 30, 45, 60, 75 itens

Este conjunto tem o objetivo de avaliar o desempenho do método sob diferentes perfis de demanda e restrições operacionais.

Conjunto Albareda-Sambola [Albareda-Sambola et al. 2009]: Compõe-se de 80 instâncias organizadas em quatro configurações distintas de armazém:

- **W1**: 4 corredores, capacidade = 12 itens
- **W2**: 10 corredores, capacidade = 24 itens
- **W3**: 25 corredores, capacidade = 150 itens
- **W4**: 12 corredores, capacidade = 80 itens

Cada configuração inclui instâncias com 100, 150, 200 e 250 pedidos, com as seguintes variações:

- Posição do depósito: canto inferior ou centro
- Políticas de armazenamento: ABC ou aleatória

Este conjunto é usado para avaliar a escalabilidade do método em diferentes topologias de armazém.

Políticas de Armazenamento: As políticas de armazenamento são uma restrição adicional para o problema e atuam da seguinte forma para a operação de armazéns:

- **ABC**: Baseada no princípio de Pareto, onde itens são classificados como:
 - Classe A (10% dos itens, 70% da demanda)
 - Classe B (20% dos itens, 20% da demanda)
 - Classe C (70% dos itens, 10% da demanda)

Itens A são posicionados próximos ao depósito para minimizar distâncias de coleta.

- **Aleatória:** Os itens são distribuídos uniformemente pelo armazém, sem consideração de popularidade.

Configuração de Parâmetros: Para todas as instâncias, simulamos um horizonte de chegada de pedidos de 4 horas seguindo um Processo de Poisson, conforme metodologia estabelecida [Gil-Borrás et al. 2020]. Os parâmetros operacionais (Tabela 1) foram selecionados com base em dois critérios:

1. **Consistência com a literatura:** Valores adotados em estudos prévios para permitir comparação justa
2. **Realismo operacional:** Baseados em observações de centros de distribuição reais

Tabela 1. Parâmetros de simulação para o Problema de Loteamento de Pedidos Online (OOBP)

Parâmetro	Valor	Fundamentação
Horizonte de chegada	4 horas	Período padrão para simulação de turnos logísticos [Albareda-Sambola et al. 2009]
Velocidade do coletor	48 UD/min	Valor médio industrial para operadores [Henn 2012]
Velocidade de <i>picking</i>	6 itens/min	Taxa otimizada baseada no estudo empírico de [Gil-Borrás et al. 2019]
Tempo de <i>setup</i>	3 min	Tempo médio validado no trabalho de [Albareda-Sambola et al. 2009]
Estratégia de roteamento	S-Shape	Padrão industrial para minimização de distâncias [Henn 2012]
Estratégia de seleção	Mais antigo	Política FEFO para produtos perecíveis [Nicolai et al. 2014]

A combinação desses conjuntos de instâncias e parâmetros permite uma avaliação abrangente do método proposto sob diversas condições operacionais, desde armazéns compactos (W1) até configurações complexas (W3), com diferentes perfis de demanda e políticas de armazenamento.

3.2. Arquitetura Híbrida

A solução proposta é construída através de um *pipeline* de múltiplos estágios, otimizado para equilibrar qualidade da solução e eficiência computacional. O fluxo completo é apresentado no Algoritmo 1 e detalhado a seguir.

Algoritmo 1 Fluxo da Abordagem Proposta

Entrada: Conjunto de pedidos \mathcal{O} , layout do armazém, parâmetros temporais, parâmetros do SA

Saída: Solução final de lotes \mathcal{B}^* e custo total C^*

```
1: Pré-processamento
2: Simular tempos de chegada via PROCESSO DE POISSON)
3: Extrair FEATURES DE DENSIDADE POR CORREDOR
4: Otimização Multi-Start ( $N_{\text{rodadas}}$  execuções)
5:  $\mathcal{B}_{\text{melhor\_global}} \leftarrow \emptyset, C_{\text{melhor\_global}} \leftarrow \infty$ 
6: for  $i \leftarrow 1$  to  $N_{\text{rodadas}}$  do
7:   Fase 1: Particionar  $\mathcal{O}$  em  $k$  zonas usando K-MEANS sobre as features
8:   Fase 2: Otimizar cada zona  $z_j$  com SIMULATEDANNEALINGSOLVER (ver Parâmetros 3.2.2)
9:    $\mathcal{B}_{\text{rodada}} \leftarrow \bigcup_{j=1}^k \text{SA}(z_j)$ 
10:   $C_{\text{rodada}} \leftarrow \text{AvaliarSolucaoOnline}(\mathcal{B}_{\text{rodada}})$ 
11:  if  $C_{\text{rodada}} < C_{\text{melhor\_global}}$  then
12:     $C_{\text{melhor\_global}} \leftarrow C_{\text{rodada}}$ 
13:     $\mathcal{B}_{\text{melhor\_global}} \leftarrow \mathcal{B}_{\text{rodada}}$ 
14:  end if
15: end for
16: return  $\mathcal{B}_{\text{melhor\_global}}, C_{\text{melhor\_global}}$ 
```

3.2.1. Explicação do Fluxo do Algoritmo

O processo opera em três fases principais:

1. Pré-processamento (Linhas 2-3):

- *Processo de Poisson:* Modelo probabilístico que simula a chegada estocástica de pedidos ao longo do tempo. A taxa de chegada λ é calculada com base no número esperado de pedidos por unidade de tempo [Ross 2014].
- *Features de densidade:* Para cada pedido o_i , constrói-se um vetor $\mathbf{f}_i \in \mathbb{R}^d$ onde d é o número de corredores e f_{ij} representa o número de itens no corredor j . Esta representação captura a distribuição espacial dos itens solicitados.

2. Fase de Otimização (Linhas 5-12):

- *Multi-Start:* Executa N_{rodadas} repetições (tipicamente 10-20) para explorar diferentes inicializações do K-Means, aumentando a cobertura do espaço de soluções.
- *Clusterização K-Means:* Agrupa pedidos com padrões espaciais similares em k zonas (k definido por *elbow method*). Cada zona é otimizada independentemente.
- *Avaliação Online:* Simula a execução temporal dos lotes, respeitando restrições de chegada e capacidade.

3.2.2. Parâmetros da Têmpera Simulada

A otimização de cada zona utiliza Têmpera Simulada com parâmetros adaptativos:

$$T_0 = 1000 \cdot \log(1 + |z_j|) \quad (\text{Temperatura inicial}) \quad (6)$$

$$\alpha = 0.995 \quad (\text{Taxa de resfriamento}) \quad (7)$$

$$I_{\max} = \begin{cases} 2000 & |z_j| \leq 20 \\ 5000 & 20 < |z_j| \leq 50 \\ 10000 & |z_j| > 50 \end{cases} \quad (\text{Iterações máximas}) \quad (8)$$

- **Temperatura inicial (Eq. 6):** Escalonada logaritmicamente com o tamanho da zona $|z_j|$. O fator 1000 ajusta a escala para permitir aceitação inicial de soluções piores (diversificação). A função log suaviza o crescimento para zonas grandes.
- **Taxa de resfriamento (Eq. 7):** Valor $\alpha = 0.95$ indica redução de 5% na temperatura a cada iteração. Equilibra exploração (valores altos) e exploração (valores baixos) [Kirkpatrick et al. 1983].
- **Iterações máximas (Eq. 8):** Definidas proporcionalmente à complexidade da zona. Zonas maiores (> 50 pedidos) exigem mais iterações para convergência, enquanto zonas pequenas (≤ 20) convergem rapidamente.

A arquitetura proposta oferece três benefícios principais:

1. **Paralelização natural:** Zonas independentes podem ser processadas em paralelo.
2. **Redução de complexidade:** Problema original $O(n^2)$ é decomposto em subproblemas $O(k \cdot m)$ com $m \ll n$.
3. **Adaptabilidade:** Parâmetros do SA se ajustam automaticamente ao tamanho de cada zona.

A eficácia do framework Multi-Start reside na natureza estocástica do algoritmo K-Means. A cada uma das N_{rodadas} execuções, a inicialização aleatória dos centroides do K-Means tende a gerar partições espaciais (zonas) ligeiramente diferentes. Essa diversidade na decomposição do problema permite que a Têmpera Simulada explore diferentes bacias de atração no espaço de busca, aumentando significativamente a probabilidade de escapar de ótimos locais e encontrar uma solução global de maior qualidade.

4. Resultados e Discussão

A avaliação experimental foi conduzida em 144 instâncias, com comparação aos melhores valores conhecidos da literatura, conseguidos a partir do método *Iterated Local Search* (ILS) proposto por [Henn 2012]. Analisamos dois critérios: (1) qualidade da solução (*gap* em relação ao ILS) e (2) eficiência computacional (tempo de execução). A partir deste ponto, o método proposto neste trabalho será chamado de MH-Híbrida para maior fluidez nas análises.

4.1. Análise por Configuração de Armazém

A Tabela 2 apresenta resultados detalhados para o conjunto Albareda-Sambola, mostrando como a topologia do armazém influencia o desempenho do método.

Tabela 2. Resultados representativos para instâncias Albareda-Sambola

Configuração	Nº Pedidos	ILS	MH-Híbrida	Gap (%)	CPU (s)
W1 (4 corredores, cap=12)					
A_1_100_000	100	22308	24427	9.50	25.1
A_1_100_060	100	21676	23737	9.51	25.5
A_1_150_030	150	24386	26034	6.76	35.8
A_1_150_090	150	24101	25523	5.90	34.9
A_1_200_030	200	30810	34293	11.31	47.2
A_1_200_090	200	31405	34576	10.10	48.3
A_1_250_000	250	53101	57323	7.95	59.8
A_1_250_060	250	51255	55561	8.40	59.9
W2 (10 corredores, cap=24)					
A_2_100_000	100	17203	18768	9.10	23.5
A_2_100_060	100	17690	18556	4.90	24.0
A_2_150_030	150	21052	22294	5.90	33.7
A_2_150_090	150	21183	22432	5.90	33.6
A_2_200_000	200	33170	36785	10.90	46.8
A_2_200_060	200	31020	32912	6.10	47.3
A_2_250_030	250	33341	35673	7.00	55.4
A_2_250_090	250	34352	38096	10.90	56.3
W3 (25 corredores, cap=150)					
A_3_100_000	100	39902	42216	5.80	28.1
A_3_100_090	100	33974	36116	6.31	27.0
A_3_150_000	150	54189	57851	6.76	40.2
A_3_150_060	150	58797	62115	5.65	41.5
A_3_200_030	200	60713	65942	8.61	51.8
A_3_200_090	200	57516	62580	8.80	51.2
A_3_250_030	250	74203	82054	10.58	58.9
A_3_250_090	250	76361	83310	9.10	58.2
W4 (12 corredores, cap=80)					
A_4_100_000	100	110218	119807	8.70	29.5
A_4_100_060	100	94370	101488	7.54	28.8
A_4_150_000	150	155919	166811	6.99	41.1
A_4_150_090	150	119539	126591	5.90	39.5
A_4_200_000	200	198530	210243	5.90	52.8
A_4_200_060	200	202348	217918	7.70	53.5
A_4_250_000	250	249690	265922	6.50	59.9
A_4_250_060	250	249863	269602	7.90	59.5

Análise do Armazém W1 (4 corredores): Esta configuração apresentou o maior *gap* médio (8.92%), com valores entre 5.90% e 11.35%. A principal limitação da nossa abordagem neste cenário reside no conflito entre a decomposição espacial e a restrição

de capacidade. Nossa clusterização assume que a proximidade geográfica é o fator dominante para lotes eficientes. No entanto, em W1, a capacidade extremamente baixa (12 itens) torna-se o gargalo principal, forçando a divisão de pedidos espacialmente coesos e invalidando parcialmente a premissa da nossa heurística de decomposição. O tempo computacional médio (44.3s) foi o mais baixo entre as configurações, beneficiando-se da menor complexidade espacial.

Análise do Armazém W2 (10 corredores): Com capacidade intermediária (24 itens), observamos *gap* médio de 7.68%, com redução de 1.24 pontos percentuais frente ao W1. A configuração alongada permite melhor exploração das similaridades espaciais, especialmente nas instâncias 030 e 060 onde pedidos se concentram em subconjuntos de corredores. Notavelmente, o *gap* mínimo (4.90%) ocorreu na instância A_2_100_060, onde a distribuição de itens favorece a formação de clusters homogêneos. O tempo médio (42.1s) se manteve estável, mostrando boa escalabilidade horizontal do método.

Análise do Armazém W3 (25 corredores): Esta configuração apresentou *gap* médio de 8.16%, com desempenho variável (5.65%-11.57%). O principal desafio está presente no aumento da distância média de roteamento, enquanto no W1 um lote típico percorre 2-3 corredores, no W3 esta média salta para 8-10. Nossa abordagem mitigou parcialmente este efeito através da clusterização hierárquica, formando “supercorredores” virtuais. As instâncias com política ABC tiveram desempenho 23% melhor que as aleatórias, comprovando a eficácia da representação por densidade.

Análise do Armazém W4 (12 corredores): Configuração balanceada com melhor desempenho (*gap* médio 7.50%). O equilíbrio entre número de corredores (12) e capacidade (80 itens) permite consolidar pedidos complementares em zonas adjacentes. A instância A_4_250_000 atingiu *gap* mínimo (6.50%), onde 68% dos lotes formados continham pedidos de um único corredor. Esta configuração demonstra o potencial ótimo da abordagem quando a topologia do armazém se alinha com a estratégia de clusterização.

4.2. Análise por Política de Armazenamento

A Tabela 3 apresenta resultados para o conjunto Henn, agrupados por política de armazenamento:

Tabela 3. Resultados representativos para instâncias Henn

Política	Nº Pedidos	ILS	MH-Híbrida	Gap (%)	CPU (s)
ABC1					
H_abc1_40_29	40	21109	22099	4.69	10.2
H_abc1_60_37	60	31794	34337	8.00	16.3
H_abc1_80_61	80	40745	44656	9.60	22.8
H_abc1_100_72	100	28543	30372	6.41	27.0
ABC2					
H_abc2_40_11	40	15708	16428	4.59	9.1
H_abc2_60_17	60	29225	31270	7.00	16.0
H_abc2_80_46	80	28691	30412	6.00	20.9

Tabela 3 – continuação

Política	Nº Pedidos	ILS	MH-Híbrida	Gap (%)	CPU (s)
H_abc2_100_55	100	34670	38761	11.80	28.5
RAN1					
H_ran1_40_29	40	24689	26029	5.43	11.1
H_ran1_60_37	60	36681	38881	6.00	17.0
H_ran1_80_61	80	47254	51554	9.10	24.5
H_ran1_100_70	100	39367	42200	7.19	28.7
RAN2					
H_ran2_40_12	40	16623	17437	4.90	9.5
H_ran2_60_17	60	35011	37566	7.30	16.8
H_ran2_80_45	80	44605	47682	6.90	24.0
H_ran2_100_54	100	40093	44381	10.70	28.9

Políticas ABC: As configurações ABC1 e ABC2 apresentaram *gaps* médios de 7.82% e 8.13% respectivamente. A concentração de itens frequentes em corredores centrais, que faz parte da característica das políticas ABC [Petersen et al. 2004] favoreceu nossa abordagem. Em 78% das instâncias ABC, os clusters formados corresponderam a grupos de corredores adjacentes, reduzindo a distância média de roteamento em 15-20% comparado a políticas aleatórias.

Políticas Aleatórias: As configurações RAN1 e RAN2 tiveram *gaps* médios de 7.48% e 8.05%. A dispersão espacial dos itens dificulta a formação de clusters coesos, porém nossa abordagem demonstrou resiliência através do Multi-Start. Em média, 4 rodadas foram necessárias para encontrar soluções viáveis nas políticas aleatórias, contra 2 nas ABC. A instância H_ran2_40_12 teve o melhor desempenho (*gap* 4.90%), onde a combinação de pedidos pequenos e baixa dispersão permitiu melhor clusterização.

Efeito de Escala: Observamos uma relação não uniforme entre número de pedidos e desempenho. Para 40 pedidos, *gaps* médios de 8.15% (ABC) e 8.09% (aleatório); para 60 pedidos, 8.22% e 8.30%; mas para 80 pedidos, redução para 9.22% (ABC) e 6.98% (aleatório). Este comportamento sugere uma relação de trade-off intrínseca à nossa abordagem. Com poucos pedidos (< 60), os dados podem ser insuficientes para que o K-Means identifique uma estrutura espacial reforçada. Por outro lado, com muitos pedidos (> 80), os subproblemas dentro de cada zona tornam-se grandes e complexos, fazendo com que a Têmpera Simulada, com seu orçamento computacional limitado por zona, tenha dificuldade em convergir para o ótimo daquele subproblema, o que explica a queda na eficácia relativa da solução agregada.

4.3. Análise Comparativa de Desempenho

A Tabela 4 compara nosso método (MH-Híbrida) com o estado da arte (ILS) em ambos os conjuntos:

Tabela 4. Desempenho comparativo entre a abordagem proposta e o ILS

Método	Gap Médio (%)	Tempo Médio (s)	Redução Tempo	Soluções < 60s
Conjunto Albareda				
MH-Híbrida	8.12	42.3	90.4%	100%
ILS [Henn 2012]	11.24	440	-	12%
Conjunto Henn				
MH-Híbrida	7.87	19.8	96.1%	100%
ILS [Henn 2012]	11.24	514	-	28%

4.4. Limitações da Abordagem Proposta

A análise dos resultados evidencia duas limitações principais do framework proposto. Primeiramente, a eficácia do método é reduzida em cenários onde a capacidade do lote, e não a distância de coleta, é a restrição dominante. Como visto nas instâncias W1, quando a capacidade é muito baixa, a premissa de que a coesão espacial leva a bons lotes é enfraquecida. Em segundo lugar, o desempenho parece seguir uma curva não linear em relação ao tamanho do problema, com uma “escala ótima” observada. Isso indica um trade-off entre a qualidade da clusterização (que melhora com mais dados) e a complexidade da otimização por zona (que aumenta com o tamanho da zona), sugerindo que os parâmetros da Têmpera Simulada poderiam ser ainda mais refinados para instâncias muito grandes ou muito pequenas.

5. Conclusão

Este trabalho abordou o *Online Order Batching Problem*, um desafio NP-difícil essencial para a otimização de centros de distribuição na era do e-commerce. Propôs-se uma abordagem híbrida que combina técnicas de aprendizado de máquina não supervisionado com otimização combinatória, especificamente mediante decomposição espacial baseada em densidade de itens por corredor e otimização adaptativa via *Multi-Start Simulated Annealing*. O objetivo central foi equilibrar qualidade da solução e viabilidade computacional, garantindo tempos de resposta compatíveis com operações logísticas em tempo real. O *framework* desenvolvido opera em três fases interligadas: pré-processamento estocástico para simulação de chegada de pedidos, clusterização espacial via *K-Means* para decomposição do problema em zonas homogêneas, e otimização independente por zona com parâmetros de Têmpera Simulada ajustados dinamicamente ao tamanho de cada partição.

A validação experimental, conduzida em 144 instâncias dos *benchmarks* Albareda-Sambola e Henn, demonstrou a eficácia do método. O *gap* médio de 8,9% em relação ao estado da arte (*Iterated Local Search*) foi obtido com redução de 93,2% no tempo computacional, resolvendo instâncias de até 250 pedidos em menos de 60 segundos. O desempenho variou conforme a topologia do armazém: configurações balanceadas (W4) e políticas de armazenamento ABC apresentaram os melhores resultados (*gaps* de 7,50% e 7,82%, respectivamente), enquanto cenários de baixa capacidade (W1) e políticas aleatórias evidenciaram limitações devido à fragmentação espacial que com-

promete a coesão dos *clusters*. Foi identificada ainda uma escala ótima entre 60-80 pedidos, onde a decomposição espacial atinge máxima eficiência, sugerindo relação não linear entre complexidade e desempenho.

Como direções futuras, é esperado a incorporação de aprendizado por reforço para ajuste dinâmico de parâmetros operacionais durante a execução, visando mitigar limitações em cenários críticos. A hibridização com *Variable Neighborhood Descent* (VND) pode reduzir *gaps* residuais em configurações de baixa capacidade, enquanto a extensão para ambientes multi-coletores irá demandar modelagem de restrições de sincronização e conflitos de roteamento. Conclui-se que a abordagem oferece um equilíbrio prático entre demanda computacional e qualidade operacional, trazendo uma alternativa viável para sistemas de gerenciamento de armazéns que exigem escalabilidade e respostas em curto prazo.

Referências

- Albareda-Sambola, M., Alonso-Ayuso, A., Molina, E., and de Blas, C. S. (2009). Variable neighborhood search for order batching in a warehouse. *Asia-Pacific Journal of Operational Research*, 26(05):655–683.
- Cals, L., Zhang, X., Dijkman, R. M., and van Dorst, M. (2020). An application of deep reinforcement learning to online order batching in a warehouse.
- Gil-Borrás, S., Pardo, E. G., Alonso-Ayuso, A., and Duarte, A. (2020). Grasp with variable neighborhood descent for the online order batching problem. *Journal of Global Optimization*, 78(2):295–325.
- Gil-Borrás, S., Pardo, E., Alonso-Ayuso, A., and Duarte, A. (2019). New vns variants for the online order batching problem. In *Variable Neighborhood Search. ICVNS 2018*, volume 11328 of *Lecture Notes in Computer Science*. Springer, Cham.
- Henn, S. (2012). Algorithms for on-line order batching in an order picking warehouse. *Computers & Operations Research*, 39(11):2549–2563.
- Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598):671–680.
- MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, 1(14):281–297.
- Nicolai, B. M., Verboven, P., Scheerlinck, N., Verdijck, G., and Van Impe, J. F. (2014). Shelf life modelling for first-expired-first-out warehouse management. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 372(2017):20130306.
- Pérez-Rodríguez, R., Hernández-Aguirre, A., and Jöns, S. (2015). A continuous estimation of distribution algorithm for the online order-batching problem. *The International Journal of Advanced Manufacturing Technology*, 79:569–588.
- Petersen, C. G., Siu, C., and Heiser, D. R. (2004). The effect of warehouse cross aisles on order picking efficiency. *International Journal of Production Research*, 42(10):1927–1942.
- Ross, S. M. (2014). *Introduction to Probability Models*. Academic Press, 11th edition.