



BRUNO PIOVESAN MELCHIORI PERUZZA

**ALGORITMOS PARA AGRUPAMENTO DE PEDIDOS EM
LOTES EM ARMAZÉM DIVIDIDO EM ZONAS**

LAVRAS – MG

2022

BRUNO PIOVESAN MELCHIORI PERUZZA

**ALGORITMOS PARA AGRUPAMENTO DE PEDIDOS EM LOTES EM ARMAZÉM
DIVIDIDO EM ZONAS**

Dissertação apresentada à Universidade Federal de Lavras, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, para a obtenção do título de Mestre.

Prof. Dr. Dilson Lucas Pereira
Orientador

**LAVRAS – MG
2022**

Ficha catalográfica elaborada pelo Sistema de Geração de Ficha Catalográfica da Biblioteca Universitária da UFLA, com dados informados pelo(a) próprio(a) autor(a).

Peruzza, Bruno Piovesan Melchiori.

Algoritmos para agrupamento de pedidos em lotes em armazém dividido em zonas / Bruno Piovesan Melchiori Peruzza. - 2022.

65 p. : il.

Dissertação (mestrado) - Universidade Federal de Lavras, 2022.

Orientador: Prof. Dr. Dilson Lucas Pereira.

Bibliografia.

1. Coleta de pedidos. 2. Agrupamento de pedidos. 3. Lotes. 4. Zonas. 5. Heurística. 6. Otimização. 7. Modelo matemático. I. Pereira, Dilson Lucas. II. Título.

BRUNO PIOVESAN MELCHIORI PERUZZA

**ALGORITMOS PARA AGRUPAMENTO DE PEDIDOS EM LOTES EM ARMAZÉM
DIVIDIDO EM ZONAS**

ALGORITHMS FOR ORDER BATCHING IN WAREHOUSE DIVIDED INTO ZONES

Dissertação apresentada à Universidade Federal de Lavras, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, para a obtenção do título de Mestre.

APROVADA em 26 de Agosto de 2022.

Prof. Dr. Dilson Lucas Pereira	UFLA
Prof. Dr. Mayron César de Oliveira Moreira	UFLA
Prof. Dr. Cristiano Arbex Valle	UFMG

Prof. Dr. Dilson Lucas Pereira
Orientador

**LAVRAS – MG
2022**

Somente através do conhecimento podemos ser livres e justos .:

AGRADECIMENTOS

Agradeço à minha família e à minha noiva por todo o apoio antes e durante este trabalho.

Ao meu orientador, professor Dr. Dilson Lucas Pereira, por todos os direcionamentos e ensinamentos fornecidos, além de todo o acompanhamento e apoio nesta jornada.

A todos os docentes envolvidos direta ou indiretamente durante esta jornada, por me fazer um melhor profissional e contribuir para minha formação acadêmica.

Agradeço à Universidade Federal de Lavras por disponibilizar recursos para a realização do trabalho.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

RESUMO

A coleta de objetos em armazéns para atender pedidos é um processo rotineiro no comércio, sendo uma parte importante e de muito impacto nas vendas. Principalmente em compras que serão entregues posteriormente, como as on-line, a busca de objetos e a preparação do pedido podem ser bem planejados, garantindo qualidade e eficiência. Considerando um armazém com estrutura definida previamente, objetos alocados dentro dele organizados em prateleiras paralelas, gerando corredores entre elas, e uma lista de pedidos para atendimento, o foco é coletar todos os objetos de maneira a atender todos os pedidos com o menor tempo de coleta possível. Entre as estratégias para minimização de tempo de coleta, é considerado neste trabalho o agrupamento de pedidos em lotes. Neste caso específico, o armazém é dividido em zonas, cada funcionário é responsável por uma zona e a lista de pedidos é definida previamente. O objetivo deste trabalho é implementar algoritmos que resolvam o problema, utilizando os modelos matemáticos propostos e heurísticas desenvolvidas, considerando redução de tempo para coleta dos objetos, tentando alcançar uma solução ótima. Ao final, são apresentados os resultados obtidos, junto com a análise dos dados, sendo obtidos melhores resultados utilizando *Set Partitioning*.

Palavras-chave: Coleta de pedidos. Agrupamento de pedidos. Lotes. Zonas. Heurística. Modelo matemático. Otimização.

ABSTRACT

Collecting objects in warehouses to fulfill orders is a routine process in commerce. Efficient and well planned object search and order preparation contribute to a positive customer experience. Considering a warehouse with a previously defined structure, objects stored within it organized in shelves with no cross-aisles, and a list of orders to service, the focus of the problem we consider is to fulfill all orders in the shortest possible time. Among the strategies to minimize collection time, grouping orders in batches is considered in this work. In this specific case, the warehouse is divided into zones, each assigned to an employee, and the order list is defined beforehand. The objective of this work is to implement algorithms that solve the problem, using the proposed mathematical models and developed heuristics, considering the reduction of time to collect the objects, trying to reach an optimal solution. At the end, the results obtained are presented, together with the data analysis, with better results being obtained using Set Partitioning.

Keywords: Order Picking. Order batching. Batching. Zone Picking. Heuristic. Mathematical model. Optimization.

LISTA DE FIGURAS

Figura 1.1 – Armazém	15
Figura 1.2 – Local dos produtos	16
Figura 1.3 – Heurísticas de roteamento	17
Figura 1.4 – Divisão do armazém em zonas	17
Figura 2.1 – Possível layout de armazém	23
Figura 3.1 – Grafo de roteamento	32
Figura 4.1 – Heurísticas de solução - Aleatória	36
Figura 4.2 – Heurísticas de solução - Gulosa	37
Figura 4.3 – Heurísticas de solução - Gulosa ordenada	37
Figura 4.4 – Heurísticas de solução - Gulosa por similaridade de rota	38
Figura 4.5 – Heurísticas de solução - Gulosa por diferença de rota	39

LISTA DE QUADROS

Quadro 2.1 – Comparativo do referencial teórico	28
Quadro 5.1 – Resultados dos experimentos computacionais	43
Quadro A.1 – Resultados da primeira modelagem de cálculo de tempo, na primeira etapa, para zona progressiva	51
Quadro A.2 – Resultados da primeira modelagem de cálculo de tempo, na segunda etapa, para zona progressiva	52
Quadro A.3 – Resultados da segunda modelagem de cálculo de tempo, na primeira etapa, para zona progressiva	52
Quadro A.4 – Resultados da segunda modelagem de cálculo de tempo, na segunda etapa, para zona progressiva	53
Quadro A.5 – Resultados do <i>Set Partitioning</i> , na primeira etapa, para zona progressiva	53
Quadro A.6 – Resultados do <i>Set Partitioning</i> , na segunda etapa, para zona progressiva	54
Quadro A.7 – Resultados do <i>Set Partitioning</i> , na terceira etapa com instâncias de tamanho 50, para zona progressiva	54
Quadro A.8 – Resultados do <i>Set Partitioning</i> , na terceira etapa com instâncias de tamanho 100, para zona progressiva	55
Quadro A.9 – Resultados do <i>Set Partitioning</i> , na terceira etapa com instâncias de tamanho 150, para zona progressiva	55
Quadro A.10 – Resultados do <i>Set Partitioning</i> , na terceira etapa com instâncias de tamanho 200, para zona progressiva	56
Quadro A.11 – Resultados das heurísticas de solução, na primeira etapa, para zona progressiva	57
Quadro A.12 – Resultados das heurísticas de solução, na segunda etapa, para zona progressiva	58
Quadro A.13 – Resultados da primeira modelagem de cálculo de tempo, na primeira etapa, para zona sincronizada	59
Quadro A.14 – Resultados da primeira modelagem de cálculo de tempo, na segunda etapa, para zona sincronizada	59
Quadro A.15 – Resultados da segunda modelagem de cálculo de tempo, na primeira etapa, para zona sincronizada	60
Quadro A.16 – Resultados da segunda modelagem de cálculo de tempo, na segunda etapa, para zona sincronizada	60
Quadro A.17 – Resultados do <i>Set Partitioning</i> , na primeira etapa, para zona sincronizada	61
Quadro A.18 – Resultados do <i>Set Partitioning</i> , na segunda etapa, para zona sincronizada	61

Quadro A.19 –Resultados do <i>Set Partitioning</i> , na terceira etapa com instâncias de tamanho 50, para zona sincronizada	62
Quadro A.20 –Resultados do <i>Set Partitioning</i> , na terceira etapa com instâncias de tamanho 100, para zona sincronizada	62
Quadro A.21 –Resultados do <i>Set Partitioning</i> , na terceira etapa com instâncias de tamanho 150, para zona sincronizada	63
Quadro A.22 –Resultados do <i>Set Partitioning</i> , na terceira etapa com instâncias de tamanho 200, para zona sincronizada	63
Quadro A.23 –Resultados das heurísticas de solução, na primeira etapa, para zona sincronizada . .	64
Quadro A.24 –Resultados das heurísticas de solução, na segunda etapa, para zona sincronizada . .	65

LISTA DE SIGLAS

HA	Heurística aleatória
HG	Heurística gulosa
HGO	Heurística gulosa ordenada
HGS	Heurística gulosa ordenada por similaridade de rota
HGD	Heurística gulosa ordenada por diferença de rota
HBSP	Heurística baseada nos modelos de <i>Set Partitioning</i>
T	Tipo
N	Número de produtos
QP	Número total de objetos, na média
P/S	Estratégia de coleta empregada, progressiva ou sincronizada
UB	Menor limite superior
LB	Maior limite inferior

LISTA DE SÍMBOLOS

\mathcal{P}	Conjunto de pedidos
P	Um pedido
$q(i)$	Tempo para coleta dos objetos i
L	Um lote
$t^k(L)$	Tempo de coleta dos pedidos da zona k do lote L
$t(L)$	Tempo de coleta dos pedidos do lote L
C	Capacidade máxima do equipamento de coleta
z_{lk}	Momento que o lote l começa a ser processado na zona k
x_{Pl}	Variável binária, indica se o pedido P está no lote Ll
t_{lk}	Tempo de processamento do lote l na zona k
y_a^{lk}	Variável binária, indica se o arco a é utilizado para o lote l na zona k
d_a	Distância necessária para percorrer a aresta a
h^{lk}	Tempo utilizado para percorrer horizontalmente o trajeto
v^{lk}	Tempo utilizado para percorrer verticalmente o trajeto
w^{lk}	Auxiliar para garantir um total par de y na solução

SUMÁRIO

1	INTRODUCAO	15
1.1	Definição do problema neste trabalho	18
1.2	Definição formal do problema	20
1.2.1	Estratégia progressiva	20
1.2.2	Estratégia sincronizada	21
1.3	Objetivos	21
1.4	Justificativa e lacuna de pesquisa	21
1.5	Organização do trabalho	21
2	REFERENCIAL TEÓRICO	22
2.1	Economia	22
2.2	Layout do armazém	22
2.3	Algoritmos de roteamento	24
2.3.1	S-shape	24
2.3.2	Largest-gap	24
2.3.3	Return	24
2.3.4	Midpoint	24
2.4	Custo computacional	25
2.5	Divisão do armazém em zonas	25
2.6	Agrupamento de pedidos	25
2.7	Trabalhos relacionados	26
3	MODELOS DE PROGRAMAÇÃO INTEIRA MISTA	29
3.1	Modelos Compactos	29
3.1.1	Zonas progressivas	29
3.1.2	Zonas sincronizadas	30
3.1.3	Modelagem dos tempos de processamento dos lotes	31
3.1.3.1	Processamento de trajeto	31
3.1.3.2	Processamento de distância até o último corredor	32
3.2	Modelos Baseados em <i>Set Partitioning</i>	34
3.2.1	Zonas progressivas	34
3.2.2	Zonas sincronizadas	35
3.2.3	Utilização dos Modelos Baseados em <i>Set Partitioning</i>	35
4	HEURÍSTICAS	36

4.1	Heurística Aleatória	36
4.2	Heurística Gulosa	36
4.3	Heurística Gulosa Ordenada	37
4.4	Heurística Gulosa por Similaridade de Rota	37
4.5	Gulosa por diferença de rota	38
4.6	Heurísticas Baseadas nos Modelos de <i>Set Partitioning</i>	38
5	EXPERIMENTOS COMPUTACIONAIS	40
5.1	Instâncias	40
5.2	Resultados	40
6	CONCLUSÃO	44
	REFERÊNCIAS	45
A	RESULTADOS DETALHADOS	48
A.1	Primeira etapa de testes	48
A.2	Segunda etapa de testes	49
A.3	Resultados detalhados	50

1 INTRODUCAO

A evolução tecnológica e da internet possibilitou grandes avanços na comunicação entre grandes distâncias. A utilização da internet em estratégias de marketing independe do tamanho do empreendimento, como apresentado em (IVANOV, 2012). Com isto, o comércio também evoluiu, conseguindo realizar vendas e atender demandas de regiões distantes, surgindo assim empresas com estoques grandes, mantidos em armazéns. A Figura 1.1 apresenta um exemplo de armazém.

A pandemia provocada pelo COVID-19 gerou mudanças no comércio. Houve um aumento das vendas on-line, seja por opção do cliente para evitar contato ou por restrições governamentais. Isso impacta diretamente na eficiência necessária na montagem dos pedidos, devido ao grande aumento do volume de compras.

Restrições em vendas presenciais de grandes centros de distribuição, como supermercados, são exemplos claros desse impacto. Apesar de atenderem por delivery, o maior fluxo de compras é presencial, e muitas vezes não possuem uma estratégia para coleta e montagem dos pedidos que serão entregues.

Figura 1.1 – Armazém



Fonte: Reproduzida de (BOYSEN; KOSTER; WEIDINGER, 2019), com permissão da Elsevier

Diversos fatores impactam na coleta dos produtos para montagem dos pedidos, como a localização dos objetos, organização e disposição do armazém, divisão em zonas, processo de automação da coleta, caminhos a serem percorridos pelos funcionários, algoritmo de roteamento, entre outros.

A montagem do pedido ocupa grande parte do custo do processo de vendas e etapas onde o processo pode ser melhorado. Estudos relacionados a cada abordagem para melhoria da coleta e montagem são apresentados em (GILS et al., 2018), mostrando um foco mais recente em abordagens como agrupamento de pedidos em lotes e divisão do armazém em zonas.

Estudos de caso de 40 empresas são apresentados e analisados em (MARCHET; MELACINI; PEROTTI, 2014), mostrando que há um interesse na automação parcial ou total do processo. Motivadores comuns para a automação são redução de custos e melhoria da qualidade do serviço. A automação gera

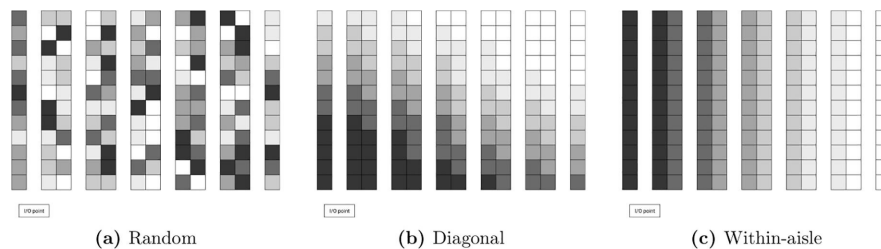
impactos para a empresa e pode ser uma transição complexa e cara. A maneira como algumas empresas realizaram o processo e os fatores mais relevantes para o seu sucesso são apresentados em (BAKER; HALIM, 2007).

O processo de coleta dos objetos do armazém para compor o pedido do cliente possui diversos pontos para enfoque de melhoria. Apesar da diversidade de contexto entre as empresas, como quantidade de funcionários, estrutura do armazém, formato e tamanho dos objetos, as soluções podem ser adaptadas a cada realidade.

A definição de melhor local para cada produto impacta diretamente no tempo de coleta. Definições de estratégias para melhor disposição dos objetos são discutidas em (SILVA et al., 2020), mostrando que a aplicação de heurísticas para a decisão melhora os resultados de políticas padrões de estabelecimentos.

A Figura 1.2 apresenta diferentes estratégias de alocação de produtos, sendo a primeira totalmente aleatória, e as demais considerando similaridade de demanda. Quanto mais escuro o quadrado, maior a quantidade de demandas, dos produtos daquele local.

Figura 1.2 – Local dos produtos



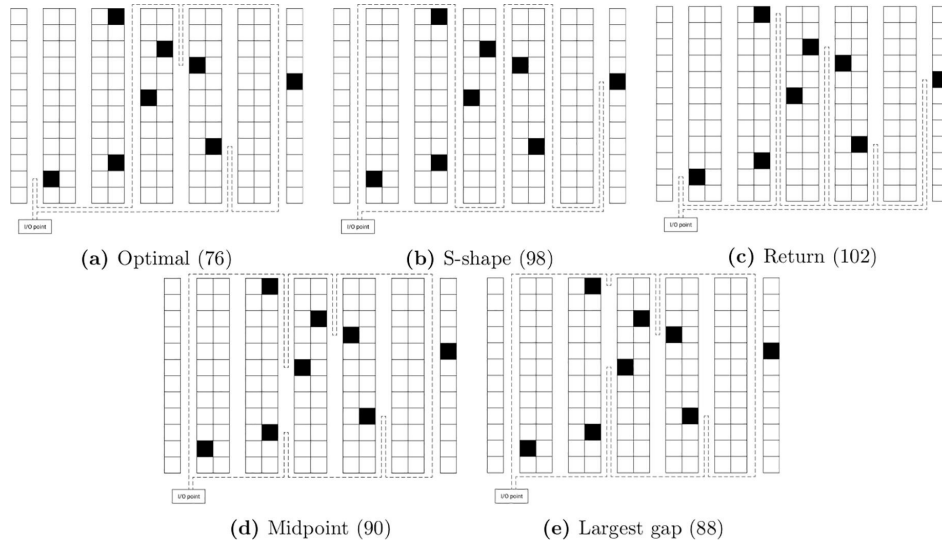
Fonte: Reproduzida de (SILVA et al., 2020), com permissão da Elsevier

Os resultados das diversas abordagens dependem da rota que será percorrida pelos funcionários, sendo este outro problema presente na coleta de pedidos. Heurísticas como S-shape, Largest-gap, Return e Midpoint são utilizadas em (HENN; WÄSCHER, 2012) e (SILVA et al., 2020) para validação da qualidade da solução.

A Figura 1.3 demonstra as heurísticas de roteamento citadas, onde os quadrados pretos são os objetos a serem coletados e a linha tracejada é a rota a ser percorrida. As heurísticas de roteamento serão detalhadas na Seção 2.3. Ao lado do nome das heurísticas está o tempo gasto para percorrer o armazém nos exemplos.

É comum que, simultaneamente, diversos funcionários colem os pedidos a eles alocados. Cada um trabalha para finalizar sua própria lista de pedidos, o que pode causar conflito entre as rotas, com situações onde dois funcionários não consigam atravessar um corredor ao mesmo tempo, e reduzindo a produtividade, além de precisarem conhecer o armazém em sua totalidade.

Figura 1.3 – Heurísticas de roteamento



Fonte: Reproduzida de (SILVA et al., 2020), com permissão da Elsevier

Dividir o armazém em zonas e cada funcionário é responsável por uma zona, organizando-os de maneira que colem colaborativamente os pedidos, é proposto em (PAN et al., 2015) e (DALLARI; MARCHET; MELACINI, 2008), mostrando que há um ganho considerável. A Figura 1.4 mostra um exemplo de divisão por zonas.

Figura 1.4 – Divisão do armazém em zonas



Fonte: Elaborada pelo autor (2022)

É comum que os funcionários percorram o armazém com um equipamento para armazenar os pedidos coletados. Este equipamento possui capacidade limitada, e portanto é necessário que se dividam

os pedidos em grupos, a serem coletados em cada viagem. Estes grupos são denominados lotes ou *batches*.

Com o objetivo de aproveitar ao máximo o trajeto percorrido por cada funcionário, realizar o agrupamento de pedidos que têm objetos próximos, de acordo com a rota que será percorrida, pode diminuir o tempo total de coleta. Modelos e heurísticas para agrupamento de pedidos são apresentados em (HENN; WÄSCHER, 2012). (VALLE; BEASLEY; CUNHA, 2017) resolvem o problema de agrupamento e de roteamento de forma integrada. Resultados promissores na redução do tempo de coleta são apresentados nesses trabalhos.

Todos os problemas citados podem ser resolvidos independentemente ou combinações podem ser consideradas. Uma visão geral sobre tais combinações é demonstrada em (BOYSEN; KOSTER; WEIDINGER, 2019) e (GILS et al., 2018), mostrando que a combinação é uma estratégia positiva para melhoria da coleta de pedidos.

1.1 Definição do problema neste trabalho

Há uma lista de pedidos a serem coletados em um armazém. Cada pedido é composto por diferentes objetos. Funcionários recebem uma lista de objetos que devem ser coletados durante seu trajeto pelo armazém. Partindo de um local inicial, ele trafega pelo armazém para coletar os objetos com uma estratégia de roteamento pré-definida.

Ele tem posse ou controle de um equipamento de transporte para carregar os objetos, o qual é restrito por um limite de capacidade, sendo que neste caso o limite é relacionado ao peso ou tamanho dos objetos.

O layout do armazém considerado é de prateleiras paralelas, gerando corredores entre elas. Cada produto e todas as unidades dele estão presentes em apenas um local do armazém, não sendo possível encontrar um mesmo produto em partes diferentes do armazém.

Devido a tal limitação, não é possível que o funcionário realize a coleta de todos os pedidos simultaneamente, sendo necessário realizar a divisão dos pedidos para que sejam coletados separadamente, o que é chamado de loteamento ou *batching*.

Separar os objetos de um pedido, em diferentes coletas, pode aumentar o custo, pois será necessária uma etapa de organização geral dos pedidos após a coleta, estando também sujeita a erros. Sendo assim, todos os objetos de um pedido devem ser recolhidos durante um mesmo trajeto, alocando-os em um mesmo lote.

Cada pedido é composto por diversos objetos, o armazém onde estão os objetos será dividido em zonas, de forma que um funcionário ficará responsável por apenas uma, o objetivo é determinar, a partir

de uma lista de pedidos, a melhor criação de lotes de pedidos para serem coletados em conjunto, visando minimizar o tempo da coleta.

Será aplicada a divisão em zonas devido aos benefícios gerados na coleta, para evitar conflito de trajeto entre os funcionários e aumentar a produtividade. A divisão é algo possível em contextos reais, como citados e exemplificados em (HO; LIN, 2017).

O processo de adaptação é de fácil aprendizado pelos funcionários, visto que ficarão com uma quantidade reduzida de corredores para percorrer. Com isto, os objetos de um pedido podem ser coletados por funcionários diferentes. Cada zona tem seu próprio ponto de partida definido, que é o início do primeiro corredor daquela zona.

No atual problema, é considerado que a lista de pedidos é totalmente conhecida e não será alterada. A divisão do armazém em zonas já existirá, sendo cada zona alocada a apenas um coletor. Devido a divisão em zonas, é necessário definir a estratégia para coleta de objetos, que pode ser de maneira progressiva ou sincronizada, sendo ambas consideradas neste trabalho.

Na abordagem progressiva, dado um lote, o funcionário é responsável pela coleta de todos os objetos presentes em sua zona. Após isso, ele encaminha o equipamento de transporte de objetos para a próxima zona, e então poderá iniciar a coleta do próximo lote com um novo equipamento de transporte, vindo da zona anterior ou do início da rota.

Devido a possível sobrecarga de demanda em certas zonas, gerando um gargalo na coleta, é possível que as zonas sofram alterações de organização. Para este problema, não consideramos a adaptação de zonas, assumindo que serão definidas previamente, e permanecerão fixas durante toda a definição e coleta dos lotes.

Na abordagem sincronizada, dado um lote, os funcionários realizam a coleta simultaneamente dos objetos daquele lote, sendo necessário que todos terminem sua coleta antes do próximo lote iniciar. Após a coleta, todos encaminham seu equipamento de transporte para um mesmo ponto de concentração dos objetos.

Esta abordagem é preferencial quando não é possível carregar muitos objetos de uma só vez com o equipamento de transporte. É importante considerar o balanceamento das zonas no momento de criação dos lotes, visto que o tempo total é prejudicado caso uma zona demore mais tempo que as demais.

Devido a divisão em zonas, apesar de um pedido estar totalmente contido em um lote, é possível que diversos funcionários colem os objetos de um mesmo pedido. O tempo de separação dos objetos em pedidos do lote é ignorado neste trabalho.

Como o foco deste trabalho é definir o melhor agrupamento de pedidos, definimos que o roteamento utilizado é sempre o S-shape, devido a sua facilidade de implementação.

1.2 Definição formal do problema

Considera-se um grafo $G = (V, E)$, que representa o armazém. O grafo G é composto de K componentes conexas $G^k = (V^k, E^k)$, $1 \leq k \leq K$, cada uma representando uma zona do armazém, sob responsabilidade de um único coletor.

Tem-se um conjunto \mathcal{P} de pedidos. Cada pedido $P \in \mathcal{P}$ é um conjunto de objetos. Para cada objeto $i \in P$, $q(i) \in \mathbb{N}$ representa o tempo total para coleta dos objetos i , sendo a quantidade solicitada daquele objeto multiplicada pelo tempo de coleta de cada objeto, $v(i) \in V$ representa o local (vértice) do armazém onde aquele objeto está localizado, e $c(i) \in \mathbb{R}_+$ representa o quanto o total solicitado daquele objeto ocupa da capacidade do coletor.

Um lote $L \subseteq \mathcal{P}$ é um subconjunto de pedidos. O conjunto de pedidos \mathcal{P} deve ser particionado em um conjunto de lotes $\mathcal{L} = (L_1, L_2, \dots, L_{|\mathcal{L}|})$. Note que o número máximo N de lotes se dá quando cada lote tem um único pedido, $N = |\mathcal{P}|$.

Dado um subconjunto de pedidos $L \subseteq \mathcal{P}$, seja $I(L) = \cup_{P \in L} P$ o conjunto dos objetos nos pedidos do lote. Seja $I^k(L) = \{i \in I(L) : v(i) \in V^k\}$ o conjunto dos objetos do lote que se encontram na zona k . Seja $t^k(L)$ o tempo para coletar os pedidos de $I^k(L)$ na zona k de acordo com alguma estratégia de roteamento escolhida. Para o caso de um único pedido P usaremos $I(P)$ e $I^k(P)$ ao invés de $I(\{P\})$ e $I^k(\{P\})$.

1.2.1 Estratégia progressiva

Na estratégia progressiva, o equipamento de coleta utilizado em uma zona é passado para a zona seguinte. Portanto, a capacidade total ocupada pelos objetos de cada lote L não deve exceder um total C que representa a capacidade do equipamento, isto é, $\sum_{i \in I(L)} c(i) \leq C$.

Seja z_{lk} o momento em que o lote L_l , $1 \leq l \leq |\mathcal{L}|$, começa a ser processado na zona k . Assuma $z_{11} = 0$. Para a primeira zona, temos que $z_{l1} \geq z_{(l-1)1} + t^1(L_l)$, $1 < l \leq |\mathcal{L}|$, isto é, um lote só pode ser processado após o término do anterior. Para as demais zonas, temos que o primeiro lote só pode começar a ser processado após ter sido processado na zona anterior $z_{1k} \geq z_{1(k-1)} + t^{k-1}(L_1)$, e para os demais lotes $z_{lk} \geq \max\{z_{(l-1)k} + t^k(L_{l-1}), z_{l(k-1)} + t^{k-1}(L_l)\}$, isto é, um lote só pode começar a ser processado após o término do lote anterior e após ter sido liberado pela zona anterior. O objetivo é minimizar o término do último lote na última zona, isto é $z_{|\mathcal{L}|K} + t^K(L_{|\mathcal{L}|})$.

Após coletar os objetos, o funcionário retorna ao seu ponto de origem. Neste momento, o equipamento de coleta é transferido para o ponto de origem da zona seguinte. O tempo para esta transferência não é considerado neste trabalho.

1.2.2 Estratégia sincronizada

Na estratégia sincronizada, todos os pedidos no lote L são coletados simultânea e individualmente pelos coletores. Desta forma, a capacidade total ocupada pelos objetos do lote em uma zona específica k é que não deve exceder o total C , isto é, $\sum_{i \in I^k(L)} c(i) \leq C$.

Como o lote L_l , $1 < l \leq |\mathcal{L}|$, só pode ser processado após todos os coletores terminarem o processamento do lote anterior, teremos que o primeiro lote se iniciará no momento zero, $z_{1k} = 0$ para toda zona k , e os demais lotes iniciam com o maior valor de término entre todas as zonas do lote anterior, $z_{lk} \geq \max_{1 \leq k' \leq K} \{z_{(l-1)k'} + t^{k'}(L_{l-1})\}$, $1 < l \leq |\mathcal{L}|$. O objetivo é minimizar o término do último lote, isto é, minimizar $\max_{1 \leq k \leq K} \{z_{|\mathcal{L}|k} + t^k(L_{|\mathcal{L}|})\}$.

1.3 Objetivos

Este trabalho tem como objetivo o desenvolvimento de algoritmos para resolução do problema, os quais devem buscar determinar o melhor agrupamento de pedidos de maneira a otimizar o tempo da coleta.

Os objetivos específicos do trabalho são:

- a) Definir o problema;
- b) Definir as instâncias
- c) Formular modelos e heurísticas;
- d) Obter e analisar resultados dos modelos e heurísticas com as instâncias;

1.4 Justificativa e lacuna de pesquisa

Diversos algoritmos já foram desenvolvidos para situações que não consideram a divisão em zonas, que resultam em problemas de roteamento, empacotamento, entre outros. Neste trabalho é considerada a abordagem de agrupamento de pedidos em um armazém dividido em zonas.

Não encontramos qualquer trabalho na literatura que lide com o agrupamento de pedidos para o caso em que o armazém dividido em zonas.

1.5 Organização do trabalho

No Capítulo 2 apresentamos o referencial teórico e comparações com outros trabalhos, Capítulo 3 os modelos, Capítulo 4 heurísticas, Capítulo 5 os experimentos computacionais e resultados obtidos, e no Capítulo 6 uma conclusão sobre o trabalho.

2 REFERENCIAL TEÓRICO

O tema apresentado envolve áreas além da computação, como economia e matemática. Porém, ainda é pouco explorado na literatura com soluções computacionais. Nosso levantamento de artigos, até o momento de escrita deste documento, sugere que o problema é pouco explorado.

Como trabalhos relacionados, serão apresentadas abordagens computacionais para soluções de alguns problemas no processo de coleta de objetos com estruturas diversas. Como referencial teórico, será apresentada uma literatura ampla que inclui layout de armazém, estratégias de coletas dos objetos e pesquisas sobre o estado da arte na coleta de objetos.

2.1 Economia

Um grande motivador para melhoria do processo de uma empresa é a sua economia. Como a automação de tomadas de decisão e melhoria separadamente de partes de um processo podem gerar uma redução de custos, estudos sobre economia estão diretamente ligados ao presente trabalho.

A inclusão de processo automatizados e planejados em uma empresa já existente, além de casos de sucesso e dificuldades encontradas são apresentados em (MARCHET; MELACINI; PEROTTI, 2014) e (BAKER; HALIM, 2007). A adaptação de uma rotina já existente para inserir o agrupamento de pedidos em lotes e divisão em zonas pode seguir metodologias similares.

Trabalhos como (YU; KOSTER, 2009) mostram como a aplicação do agrupamento de pedidos em lotes e a divisão em zonas podem ser proveitosas, mesmo com soluções simples e suscetíveis a erros. Impactos no tempo de entrega relacionado às políticas de estoque são analisados em (LI, 2020), considerando agrupamento de pedidos.

2.2 Layout do armazém

A disposição e organização do armazém impacta diretamente no custo e no tempo para coleta dos objetos. Assim como apresentado em (QUADER; CASTILLO-VILLAR, 2018), é possível melhorar o tempo necessário para coleta dos objetos adequando a disposição e tamanho das prateleiras.

Outras possibilidades também são abordadas, como agrupar os objetos por classificação de similaridade, para então alocar os objetos similares na mesma prateleira, ou em posições próximas de prateleiras diferentes, depende do método de roteamento que será usado.

A utilização de similaridade para melhor organização dos objetos é um conceito da mineração de dados que também é apresentado em (HSIEH; HUANG, 2011), mostrando que heurísticas podem ser criadas ou otimizadas considerando a disposição do armazém.

Em situações onde o layout pode comportar prateleiras paralelas, é demonstrado em (AMARAL, 2013) que há um ganho significativo quanto a estruturas com apenas uma fileira, considerando tamanho dos objetos e das prateleiras para realização dos cálculos. É comum as propostas utilizarem estruturas paralelas como nas Figuras 1.2 e 1.3

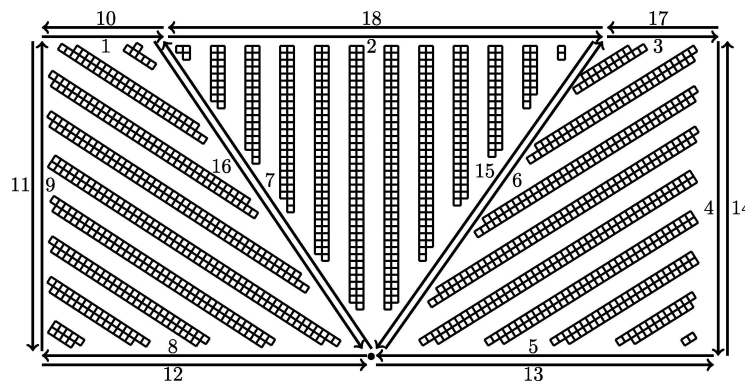
Existem estruturas onde os objetos de reserva ficam alocados em prateleiras mais altas, sendo necessário um maquinário para transferi-los para prateleiras mais baixas, para que seja possível que o funcionário acesse o objeto. O melhor tamanho do conjunto de objetos e a estrutura de alocação são discutidos em (CALZAVARA et al., 2017), considerando a condição de esforço humano, como fatores ergonômicos, e questões rentáveis para a empresa. Discussões com foco no manuseio dos objetos são apresentadas em (EKREN; SARI; LERHER, 2015).

Para estruturas paralelas, é necessário definir a profundidade das prateleiras, número de prateleiras, comprimento e largura de um armazém. Um algoritmo é apresentado em (RAKESH; ADIL, 2015) para calcular o melhor layout para reduzir os custos da coleta de objetos.

Para layouts que possuem corredores transversais é necessário adaptar o roteamento. Para tal layout, (VALLE; BEASLEY, 2020) utiliza de aproximações para definir o melhor agrupamento de pedidos em lotes, considerando a posição de seus objetos no armazém, para que seja definida a melhor rota para cada lote.

Pesquisas recentes mostram que outros layouts de prateleiras, utilizando angulações diferentes, como descritos em (OZDEN; SMITH; GUE, 2021), conseguem melhorar o desempenho da coleta de pedidos. A Figura 2.1 apresenta uma proposta de layout base, que é adaptada de acordo com o histórico de demandas dos objetos.

Figura 2.1 – Possível layout de armazém



Fonte: Reproduzida de (OZDEN; SMITH; GUE, 2021), com permissão da Elsevier

2.3 Algoritmos de roteamento

A escolha da heurística de roteamento que será utilizada impacta diretamente no agrupamento de pedidos. Ao definir qual rota será percorrida pelo funcionário, é possível definir a proximidade dos objetos, informação necessária para agrupar os pedidos. Ao alterar a heurística de roteamento, também é alterada a qualidade do agrupamento definido.

Cálculos para obter a distância média percorrida no armazém de maneira exata, de acordo com as heurísticas de roteamento S-shape (também conhecido como Traversal), Largest-gap, Return e Midpoint, são apresentados em (DIJKSTRA; ROODBERGEN, 2017). As heurísticas citadas são detalhadas em (MASAE; GLOCK; GROSSE, 2020), sendo explicadas a seguir de maneira resumida e ilustradas na Figura 1.3.

2.3.1 S-shape

A partir do ponto de partida, o funcionário se move para o corredor mais próximo que contém objetos para coleta e atravessa o corredor completamente. Ele realiza o processo repetidamente até que todos os objetos sejam coletados.

2.3.2 Largest-gap

O armazém é dividido em duas metades, partes frontal e traseira do armazém. O ponto de divisão entre as duas partes é definido para cada corredor, considerando a maior lacuna entre dois objetos ou entre um objeto e as entradas do corredor.

O funcionário entra nos corredores da parte frontal do armazém que contém objetos para coleta, deixando o corredor do lado onde ele entrou, sem acessar a parte traseira. Após passar por toda a parte frontal, ele se move para a parte traseira do armazém, com processo análogo à parte frontal.

2.3.3 Return

O funcionário entra em cada corredor que contém pelo menos um objeto para coleta, caminhando até onde é necessário para coletar todos os objetos naquele local, deixando o corredor do lado onde ele entrou. Repete-se o processo para os demais corredores.

2.3.4 Midpoint

O armazém é dividido em duas metades iguais, partes frontal e traseira do armazém. O funcionário entra nos corredores da parte frontal do armazém que contém objetos para coleta, deixando o

corredor do lado onde ele entrou, sem acessar a parte traseira. Após passar por toda a parte frontal, ele se move para a parte traseira do armazém, com processo análogo à parte frontal.

2.4 Custo computacional

Um desafio para resolução de problemas de coleta de objetos é o custo computacional necessário para execução dos algoritmos. No levantamento feito em (GILS et al., 2018), são listados diversos algoritmos, sendo necessário adaptar o modelo descrito ou preparar previamente os dados para executá-los.

A utilização de heurísticas para obtenção de uma solução em tempo viável devido ao custo computacional é recorrente na literatura, como exemplo (HENN; WÄSCHER, 2012) que, apesar de criar um algoritmo exato, o custo computacional o tornou inviável para execuções práticas.

Algumas soluções viáveis e mais genéricas conseguem destaque mesmo não conseguindo encontrar a solução ótima, como é o caso do *framework* apresentado em (CHEN et al., 2010), que aponta um direcionamento para uma boa solução do problema.

2.5 Divisão do armazém em zonas

A divisão em zonas pode considerar diversos fatores estruturais. Entre eles estão a posição dos objetos, posição das prateleiras e corredores, melhor quantidade de zonas, entre outros. Fatores estratégicos também devem ser considerados, como abordagens simultâneas ou lineares, zonas estáticas ou dinâmicas.

Alguns exemplos de divisão por zonas em (JEWKES; LEE; VICKSON, 2004) mostram a melhor alocação dos produtos no armazém e melhor divisão de zonas, (KOSTER; LE-DUC; ZAERPOUR, 2012) o número de zonas ótimo para dividir o armazém, e (JANE; LAIH, 2005) uma heurística para zonas sincronizadas.

2.6 Agrupamento de pedidos

O objetivo do agrupamento de pedidos pode diferir de acordo com o problema, podendo ser uma melhor divisão de trabalho entre os coletores, mais velocidade de coleta dos objetos para montagem do pedido, entre outros. Algumas restrições, como ordem de chegada dos pedidos e tamanho máximo para agrupamento dos pedidos, podem ser adaptadas de acordo com o contexto.

Em (HENN; WÄSCHER, 2012) são vistos alguns exemplos de agrupamento de pedidos, com diversos algoritmos propostos e prova que a redução dos custos em zonas individuais implica em uma

redução do custo total. Soluções para encontrar o tamanho ótimo de um agrupamento de pedidos em (NIEUWENHUYSE; KOSTER, 2009).

Abordagens para definir simultaneamente o melhor agrupamento de pedidos em lotes e o melhor roteamento para o lote são apresentadas em (VALLE; BEASLEY; CUNHA, 2017), resolve de maneira ótima ambos os problemas até uma certa quantidade de pedidos. Com o aumento do tamanho é aplicada uma heurística para o agrupamento, porém o roteamento continua sendo resolvido de maneira ótima.

Dois heurísticas, Tabu Search e Attribute-based Hill Climber, utilizadas para definir o melhor agrupamento de pedidos, são propostas em (HENN; WÄSCHER, 2012), conseguindo bons resultados para as técnicas de roteamento S-Shape e Largest-gap.

Pesquisas mais recentes consideram que pedidos podem ser agrupados, como em (PAN; SHIH; WU, 2015), que busca definir a menor quantidade de agrupamentos possível, mantendo as zonas balanceadas quanto ao tempo. Além disso, (HUANG et al., 2018) busca definir o melhor agrupamento dos pedidos mantendo o balanceamento, descrevendo os impactos para as abordagens progressiva e sincronizada.

2.7 Trabalhos relacionados

O problema de coleta de objetos possui variações de estrutura e diversos objetivos de melhoria. A aplicação simultânea de diversos pode gerar um resultado melhor com relação a aplicação separadamente, assim como apresentado por (GILS et al., 2018), o qual também mostra diversas abordagens já existentes para o problema de maneira geral.

Soluções para tamanhos ótimos de agrupamentos ou alocação ótima de esforço para o caso específico de armazém dividido em 2 blocos são apresentados em (NIEUWENHUYSE; KOSTER, 2009). No trabalho, é criado um modelo que pode ser utilizado para solução, sendo possível sua adaptação para políticas específicas. O trabalho deixa espaços para buscas simultâneas de pedidos por ser necessário considerar o tempo de espera dos funcionários durante o processo.

A definição da quantidade de zonas é um ponto importante para o presente trabalho. O modelo apresentado em (KOSTER; LE-DUC; ZAERPOUR, 2012) determina a melhor organização do armazém, considerando a adequação do armazém em zonas, distribuição das zonas para os funcionários e esforço necessário para cada zona.

O desempenho modelo apresentado depende da quantidade de objetos, tamanho dos pedidos e quantidade de zonas, o qual pode ser computacionalmente inviável dependendo da entrada. São apresentados testes realizados com dados reais de um centro de distribuição, mostrando assim uma solução satisfatória. O trabalho não aborda congestionamento de zonas no momento da coleta dos objetos.

Em armazéns com zonas progressivas, para obter a quantidade ótima de zonas onde haja o menor desbalanceamento de trabalho, (PAN; SHIH; WU, 2015) utilizam algumas abordagens para solução do problema. Agrupar os pedidos por ordem de recebimento não é a melhor opção.

Neste caso, se a ordem dos pedidos for irrelevante, é possível utilizar algoritmos de empacotamento para conseguir a quantidade ótima de grupos para divisão dos pedidos, porém podem gerar desbalanceamento entre as zonas. Com isto, é proposta uma heurística baseada em algoritmo genético de agrupamento, o qual consegue melhores resultados para o problema em questão.

Para coleta de lotes em armazém sem divisão por zonas, com roteamento como *S-shape* e *largest-gap*, (HENN; WÄSCHER, 2012) define o melhor agrupamento de pedidos em lotes, com soluções de complexidade exponencial, com sua execução potencialmente inviável.

Para resolver o problema são apresentadas duas meta-heurísticas, a clássica busca Tabu e o *Hill climber* baseado em atributos. Ambas dependem de soluções iniciais, com algumas abordagens apresentadas no trabalho. As meta-heurísticas são comparadas com demais abordagens da literatura, provando assim serem boas soluções, deixando ainda espaço para divisão de armazém em zonas e melhor alocação dos objetos no armazém.

A redução do tempo total de coleta pode ser obtida ao melhor dispor os objetos dentro do armazém. Considerando zonas sincronizadas, a melhor disposição dos objetos pode balancear o tempo necessário para coleta de objetos em cada zona. Um modelo de agrupamento natural é apresentado em (JANE; LAIH, 2005), o qual é resolvido através de uma heurística que utiliza informações dos pedidos.

Após comparação com o cenário atual de uma empresa, é mostrado que os resultados gerados pela solução proposta são melhores. Para a solução proposta, caso haja uma redução no tamanho das zonas, é possível melhorar o resultado, porém aumentando o esforço necessário para cada funcionário, sendo possível escolher um tamanho ótimo para as zonas de acordo com o esforço adequado para os funcionários.

Para agrupamento de pedidos em um armazém que é separado em zonas, buscando um menor desbalanceamento de trabalho por zona, (HUANG et al., 2018) propõe modelos referentes às abordagens progressiva e sincronizada. O processo apresentado aborda a coleta, separação e empacotamento dos pedidos. É demonstrado que, principalmente para zonas progressivas, há um impacto na redução do tempo de coleta ao balancear o trabalho em cada zona.

Um grande motivador para melhoria do processo de uma empresa é a sua economia. Como a automação de tomadas de decisão e melhoria separadamente de partes de um processo podem gerar uma redução de custos, estudos sobre economia estão diretamente ligados ao presente trabalho.

O processo de inclusão em uma empresa já existente, casos de sucesso e problemas encontrados são apresentados em (MARCHET; MELACINI; PEROTTI, 2014) e (BAKER; HALIM, 2007). A adaptação de uma empresa para a aplicação de agrupamento de pedidos e divisão em zonas pode seguir metodologias similares.

Trabalhos como (YU; KOSTER, 2009) mostram como a aplicação do agrupamento de pedidos em lotes e a divisão em zonas podem ser proveitosas, mesmo com soluções simples e suscetíveis a erros. Impactos no tempo de entrega relacionado às políticas de estoque são analisados em (LI, 2020), considerando agrupamento de pedidos.

Quadro 2.1 – Comparativo do referencial teórico

Este trabalho	-	✓	✓	✓	✓	✓	-
(NIEUWENHUYSE; KOSTER, 2009)	✓	-	-	✓	✓	-	-
(KOSTER; LE-DUC; ZAERPOUR, 2012)	✓	-	✓	-	-	-	-
(PAN; SHIH; WU, 2015)	-	-	✓	✓	-	-	-
(HENN; WÄSCHER, 2012)	-	✓	-	✓	✓	✓	-
(JANE; LAIH, 2005)	-	-	✓	✓	✓	-	-
(HUANG et al., 2018)	-	✓	✓	✓	✓	-	-
(VALLE; BEASLEY; CUNHA, 2017)	-	✓	-	✓	✓	✓	✓
(JEWKES; LEE; VICKSON, 2004)	-	-	✓	-	-	-	-
(CHEN et al., 2010)	-	✓	-	✓	-	-	-
(MASAE; GLOCK; GROSSE, 2020)	-	-	-	-	-	✓	-
(DIJKSTRA; ROODBERGEN, 2017)	-	-	-	-	-	✓	✓
(VALLE; BEASLEY, 2020)	-	✓	-	✓	✓	✓	✓
(OZDEN; SMITH; GUE, 2021)	✓	-	-	-	-	-	-
(RAKESH; ADIL, 2015)	✓	-	-	-	-	-	-
(EKREN; SARI; LERHER, 2015)	✓	✓	-	-	-	-	-
(CALZAVARA et al., 2017)	✓	✓	-	✓	-	-	-
(AMARAL, 2013)	✓	-	-	-	-	-	-
(HSIEH; HUANG, 2011)	✓	-	-	-	-	-	-
(QUADER; CASTILLO-VILLAR, 2018)	✓	-	-	-	-	-	-
(LI, 2020)	-	-	-	✓	-	-	-
(YU; KOSTER, 2009)	-	-	✓	✓	-	-	-
(BAKER; HALIM, 2007)	-	-	-	✓	-	-	-
(MARCHET; MELACINI; PEROTTI, 2014)	-	-	-	✓	-	-	-
Definição de layout do armazém							
Planejamento de coleta							
Divisão em zonas							
Agrupamento de pedidos							
Soluções para agrupamento							
Soluções de roteamento							
Otimização de roteamento							

Fonte: Elaborado pelo autor (2022)

O Quadro 2.1 apresenta uma comparação dos trabalhos citados neste referencial em relação a este trabalho. Apesar das similaridades, é possível perceber que os trabalhos listados não consideram os assuntos totalmente iguais a este, pois eles apresentam outros objetivos.

3 MODELOS DE PROGRAMAÇÃO INTEIRA MISTA

A seguir, são apresentados modelos de Programação Inteira Mista para o problema. É apresentado um modelo compacto e um modelo baseado em *Set Partitioning*, tanto para o caso de zonas progressivas, quanto para zonas sincronizadas.

3.1 Modelos Compactos

Os modelos compactos a seguir precisam de um mecanismo para determinar o tempo de coleta de cada lote, seguindo a heurística S-Shape. A forma como este tempo é modelado é comum a ambos os modelos e é apresentada na Seção 3.1.3.

3.1.1 Zonas progressivas

O modelo para zonas progressivas utiliza os seguintes conjuntos de variáveis de decisão:

- a) $x = \{x_{Pl} \in \{0, 1\} : P \in \mathcal{P}, 1 \leq l \leq N\}$. A variável binária x_{Pl} indica se o pedido P está no lote L_l ;
- b) $t = \{t_{lk} \in \mathbb{R}_+ : 1 \leq l \leq N, 1 \leq k \leq K\}$. A variável $t_{lk} \geq 0$ indica o tempo de processamento do lote l na zona k ;
- c) $z = \{z_{lk} \in \mathbb{R}_+ : 1 \leq l \leq N, 1 \leq k \leq K\}$. A variável $z_{lk} \geq 0$ indica o momento de início do processamento do lote l na zona k ;

O problema é modelado da seguinte forma:

$$\min z_{NK} + t_{NK} \quad (3.1)$$

Todo pedido deve estar em algum lote e apenas em um lote:

$$\sum_{l=1}^N x_{Pl} = 1, \quad P \in \mathcal{P}. \quad (3.2)$$

Restrições para o início correto dos lotes:

$$z_{11} = 0, \quad (3.3)$$

$$z_{l1} \geq z_{(l-1)1} + t_{(l-1)1}, \quad 1 < l \leq N, \quad (3.4)$$

$$z_{1k} \geq z_{1(k-1)} + t_{1(k-1)}, \quad 1 < k \leq K, \quad (3.5)$$

$$z_{lk} \geq z_{(l-1)k} + t_{(l-1)k}, \quad 1 < l \leq N, 1 < k \leq K, \quad (3.6)$$

$$z_{lk} \geq z_{l(k-1)} + t_{l(k-1)}, \quad 1 < l \leq N, 1 < k \leq K. \quad (3.7)$$

A capacidade do equipamento de coleta deve ser respeitada:

$$\sum_{P \in \mathcal{P}} \sum_{i \in P} c(i)x_{Pl} \leq C, \quad 1 \leq l \leq N. \quad (3.8)$$

Além das restrições acima, temos também as restrições de domínio das variáveis, que devem estar de acordo com definição das mesmas e serão omitidas aqui e nos modelos subsequentes.

3.1.2 Zonas sincronizadas

Além dos conjuntos de variáveis x e t , definidos anteriormente, o modelo para zonas sincronizadas utiliza uma variável $T \in \mathbb{R}_+$, que indica o término do último lote, e redefine o conjunto $z = \{z_l \in \mathbb{R}_+ : 1 \leq l \leq N\}$. A variável $z_l \geq 0$ indica o momento de início do processamento do lote l . O lote tem início simultâneo em todas as zonas.

Tem-se, então, o seguinte modelo:

$$\min T \quad (3.9)$$

Todo pedido deve estar em algum lote e apenas em um lote:

$$\sum_{l=1}^N x_{Pl} = 1, \quad P \in \mathcal{P}. \quad (3.10)$$

Restrições para o início correto dos lotes:

$$z_1 = 0, \quad (3.11)$$

$$z_l \geq z_{l-1} + t_{(l-1)k}, \quad 1 < l \leq N, 1 \leq k \leq K, \quad (3.12)$$

$$T \geq z_N + t_{Nk}, \quad 1 \leq k \leq K, \quad (3.13)$$

A capacidade do equipamento de coleta deve ser respeitada:

$$\sum_{P \in \mathcal{P}} \sum_{i \in I^k(P)} c(i)x_{Pl} \leq C, \quad 1 \leq l \leq N, 1 \leq k \leq K. \quad (3.14)$$

Além das restrições acima, têm-se também as restrições de domínio das variáveis.

3.1.3 Modelagem dos tempos de processamento dos lotes

Nos modelos, é necessário garantir que as variáveis t assumam os valores corretos, sendo apresentado nesta seção.

Assuma que a zona k do armazém está configurada com dois corredores horizontais (corredor horizontal 1 e 2) ligados por vários corredores verticais (corredor vertical $1, 2, \dots, M^k$). Assumiremos também que a heurística S-Shape será utilizada para definir os trajetos dos coletores, e que em cada zona, os coletores saem de e retornam a um mesmo ponto de origem, que se encontra na junção do corredor horizontal 1 com o corredor vertical 1.

Foram implementadas duas alternativas para modelagens de tempos, sendo uma delas processando todo o trajeto percorrido, e a outra considerando somente a distância até o último corredor de cada zona.

3.1.3.1 Processamento de trajeto

Para definir o trajeto de um lote na zona k , vamos considerar um grafo direcionado simplificado $\bar{G}^k = (\bar{V}^k, \bar{A}^k)$. Este grafo possui um vértice v_m^1 (resp. v_m^2) para a interseção de cada corredor vertical m , $1 \leq m \leq M^k$, com o corredor horizontal 1 (resp. 2).

O conjunto de arcos é constituído de dois arcos por corredor, um de v_m^1 para v_m^2 e um no sentido oposto; um arco ligando a entrada de cada corredor à entrada correspondente do corredor seguinte, isto é, de v_m^1 para v_{m+1}^1 , $1 \leq m < M^k$, no corredor 1, e arcos análogos no corredor 2; e arcos conectando $v_{M^k}^1$ e v_1^2 . A Figura 3.1 exemplifica visualmente o grafo.

Dado $a \in \bar{A}^k$, seja d_a a distância apropriada, e seja \mathcal{P}^{mk} o conjunto de pedidos que possuem algum item no corredor m da zona k . Considere variáveis binárias $\{y_a^{lk}\}$ indicando se o arco $a \in \bar{A}^k$ é utilizado na entrega do lote l na zona k .

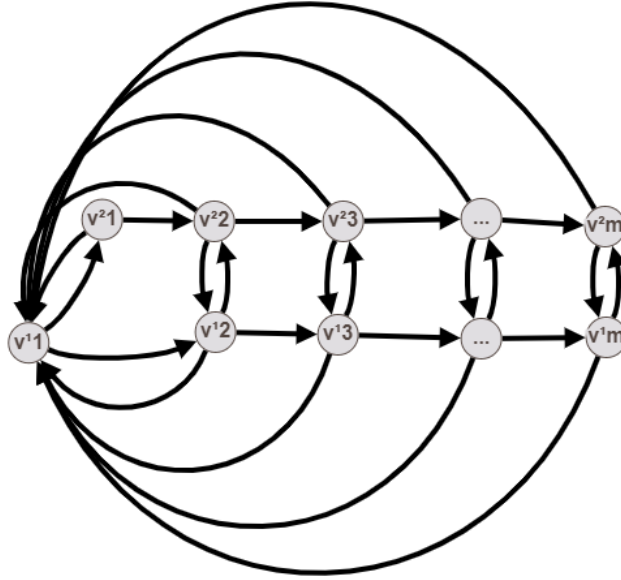
A seguinte restrição diz que se o lote l possuir algum item do corredor m da zona k , então um dos dois arcos correspondentes ao corredor deve ser percorrido:

$$y_{(v_m^1, v_m^2)}^{lk} + y_{(v_m^2, v_m^1)}^{lk} \geq x_{Pl}, \quad 1 \leq k \leq K, 1 \leq m \leq M^k, 1 \leq l \leq N, P \in \mathcal{P}^{mk}. \quad (3.15)$$

Para a coleta, um corredor só pode ser percorrido em um sentido. Para o retorno a v_1^1 , qualquer corredor poderá ser utilizado, pois a aresta de retorno não define corredores:

$$y_{(v_m^1, v_m^2)}^{lk} + y_{(v_m^2, v_m^1)}^{lk} \leq 1, \quad 1 \leq k \leq K, 1 \leq m \leq M^k, 1 \leq l \leq N. \quad (3.16)$$

Figura 3.1 – Grafo de roteamento



Fonte: Elaborada pelo autor (2022)

A próxima restrição diz que, se o coletor entrar em um vértice, então deve sair daquele vértice. Sendo assim, mesmo que o grafo simplificado possa ter ciclos que não contém a origem, a restrição garante que isso não ocorra. Portanto, a rota será sempre conexa:

$$\sum_{(u,v) \in \bar{A}^k} y_{(u,v)}^{lk} = \sum_{(v,u) \in \bar{A}^k} y_{(v,u)}^{lk}, \quad v \in \bar{V}^k, 1 \leq k \leq K, 1 \leq m \leq M^k. \quad (3.17)$$

Por fim, os tempos de deslocamento em cada zona para cada lote são dados por:

$$t_{lk} \geq \sum_{a \in \bar{A}^k} d_a y_a^{lk} + \sum_{P \in \mathcal{P}} \left(\sum_{i \in I^k(P)} q(i) \right) x_{Pl}, \quad 1 \leq l \leq N, 1 \leq k \leq K. \quad (3.18)$$

Restrições de domínio das variáveis, que devem estar de acordo com a definição.

3.1.3.2 Processamento de distância até o último corredor

Seja dv um parâmetro que indica o tempo para se percorrer um corredor vertical, e dh_{pk} um parâmetro que indica o tempo para se percorrer, o trecho do início do corredor horizontal da zona k até o início do corredor vertical mais distante utilizado na coleta do pedido P . Defina \mathcal{P}^{mk} como o conjunto de pedidos que possuem algum objeto no corredor vertical m da zona k .

A modelagem é baseada no fato de que ao se coletar um lote em uma determinada zona de acordo com a heurística S-Shape, o tempo total é dado pelo número de corredores verticais percorri-

dos multiplicado pelo tempo para se percorrer cada corredor vertical, somado ao tempo para se chegar horizontalmente ao último corredor percorrido e voltar, e ao tempo para se coletar os objetos das prateleiras. Caso o número de corredores verticais onde ocorrem coletas for ímpar, o funcionário terminará seu trajeto no corredor horizontal oposto ao ponto de partida, e precisará percorrer um corredor vertical adicional para voltar. Portanto, podemos assumir que o número de corredores percorridos verticalmente sempre será par.

Os seguintes conjuntos de variáveis de decisão serão utilizados:

- a) $y = \{y_m^{lk} \in \{0, 1\} : 1 \leq l \leq N, 1 \leq k \leq K, 0 \leq m \leq M^k\}$. A variável binária y_m^{lk} indica se o corredor vertical m , é utilizado na entrega do lote l na zona k . Note a presença de um corredor adicional 0, indicando que o funcionário terminou seu trajeto no lado oposto do ponto de origem, e precisou percorrer um corredor vertical adicional para retornar;
- b) $h = \{h^{lk} \in \mathbb{R}_+ : 1 \leq l \leq N, 1 \leq k \leq K\}$. A variável real h^{lk} indica o tempo utilizado para percorrer horizontalmente o trajeto, na coleta do lote l na zona k ;
- c) $v = \{v^{lk} \in \mathbb{R}_+ : 1 \leq l \leq N, 1 \leq k \leq K\}$. A variável real v^{lk} indica o tempo utilizado para percorrer verticalmente o trajeto, na coleta do lote l na zona k ;
- d) $w = \{w^{lk} \in \mathbb{Z}_+ : 1 \leq l \leq N, 1 \leq k \leq K\}$. A variável inteira w^{lk} é utilizada para garantir que a soma das variáveis y será par;

A seguinte restrição define que se o lote l possuir algum objeto do corredor m da zona k , então o corredor deve ser percorrido:

$$y_m^{lk} \geq x_{Pl}, \quad 1 \leq k \leq K, 1 \leq m \leq M^k, 1 \leq l \leq N, P \in \mathcal{P}^{mk}. \quad (3.19)$$

O tempo total para percorrer o trajeto horizontalmente é o dobro do tempo necessário para ir horizontalmente até o corredor mais distante nesta zona:

$$h^{lk} \geq 2dh_{P_k}x_{Pl}, \quad 1 \leq k \leq K, 1 \leq l \leq N, P \in \mathcal{P}. \quad (3.20)$$

A soma dos corredores verticais utilizados deve ser par:

$$\sum_{m=0}^{M^k} y_m^{lk} = 2w, \quad 1 \leq k \leq K, 1 \leq l \leq N. \quad (3.21)$$

Como os corredores verticais tem o mesmo tamanho, o tempo total para percorrer o trajeto verticalmente é a quantidade de corredores utilizados, multiplicada pelo tempo utilizado em cada corredor.

$$v^{lk} = dv \times \sum_{m=0}^{M^k} y_m^{lk}, \quad 1 \leq k \leq K, 1 \leq l \leq N. \quad (3.22)$$

Os tempos de deslocamento em cada zona para cada lote são dados por:

$$t_{lk} \geq h_{lk} + v_{lk} + \sum_{P \in \mathcal{P}} \left(\sum_{i \in I^k(P)} q(i) \right) x_{Pl}, \quad 1 \leq l \leq N, 1 \leq k \leq K. \quad (3.23)$$

Além das restrições de domínio das variáveis, que devem estar de acordo com a definição acima das mesmas.

3.2 Modelos Baseados em *Set Partitioning*

Os modelos abaixo consideram um conjunto \mathcal{L} de lotes pré-definidos e escolhem um subconjunto de lotes de forma que cada pedido apareça em um único lote e o tempo total de coleta seja minimizado. Para cada lote $L \in \mathcal{L}$, temos os seguintes parâmetros:

- a) $\mathcal{P}(L)$: conjunto de pedidos em L ;
- b) $t^k(L)$: tempo total para coletar todos os pedidos em L na zona k ;
- c) $t(L)$: tempo total para coletar todos os pedidos em L , i.e., $t(L) = \max_{1 \leq k \leq K} t^k(L)$;

3.2.1 Zonas progressivas

No caso de zonas progressivas, a ordem de coleta dos lotes impacta na solução. Portanto, devem ser determinados os lotes de posição 1, 2, e assim por diante, até N , no máximo. O modelo utiliza os mesmos conjuntos de variáveis t e z do modelo compacto para zonas progressivas, e introduz o conjunto $\lambda = \{\lambda_{Ll} \in \{0, 1\} : L \in \mathcal{L}, 1 \leq l \leq N\}$. A variável binária λ_{Ll} indica se o lote L foi escolhido para fazer parte da solução sendo o l -ésimo lote.

O modelo baseado em *Set Partitioning* para zonas progressivas tem a mesma função objetivo (3.1) e restrições (3.3)-(3.7) do modelo compacto para zonas progressivas, adicionadas das restrições a seguir.

Todo pedido deve estar em algum dos lotes escolhidos e apenas em um deles:

$$\sum_{L \in \mathcal{L}: P \in \mathcal{P}(L)} \sum_{l=1}^N \lambda_{Ll} = 1, \quad P \in \mathcal{P}. \quad (3.24)$$

No máximo um lote pode ser selecionado para ocupar a l -ésima posição:

$$\sum_{L \in \mathcal{L}} \lambda_{Ll} \leq 1, \quad 1 < l \leq N. \quad (3.25)$$

Atribuição do tempo em cada zona dos lotes selecionados:

$$t_{lk} = \sum_{L \in \mathcal{L}} t^k(L) \lambda_{Ll}, \quad 1 < l \leq N, 1 < k \leq K. \quad (3.26)$$

Além das restrições de domínio das variáveis, aqui omitidas.

3.2.2 Zonas sincronizadas

Uma vez que no caso de zonas sincronizadas a ordem de execução dos lotes não altera o tempo total, o modelo baseado em *Set Partitioning* fica bastante simplificado, já que não é necessário determinar o momento de início de cada lote ou a ordem dos lotes. O modelo emprega o conjunto de variáveis de decisão $\lambda = \{\lambda_L \in \{0, 1\} : L \in \mathcal{L}\}$. A variável binária λ_L indica se o lote L foi escolhido para fazer parte da solução.

Tem-se o seguinte modelo:

$$\min \sum_{L \in \mathcal{L}} t(L) \lambda_L \quad (3.27)$$

A seguinte restrição define que todo pedido deve estar em algum dos lotes escolhidos e apenas em um deles:

$$\sum_{L \in \mathcal{L} : P \in \mathcal{P}(L)} \lambda_L = 1, P \in \mathcal{P}. \quad (3.28)$$

Além das restrições de domínio das variáveis λ , que devem assumir valores 0-1.

3.2.3 Utilização dos Modelos Baseados em Set Partitioning

Para que os modelos baseados em *Set Partitioning* possam ser empregados para realizar a otimização exata do problema, é necessário que todos os possíveis lotes sejam considerados no conjunto \mathcal{L} . Tendo em vista que o número de possíveis lotes é enorme, estes não podem ser considerados de explicitamente, sendo necessário considerá-los de forma implícita, por meio de um algoritmo baseado em geração de colunas, que geraria os lotes dinamicamente, conforme o necessário. Neste trabalho, estes modelos não serão utilizados para otimização exata, mas serão utilizados como base para algoritmos heurísticos, descritos na próxima seção, que resultarão da resolução dos modelos com conjuntos restritos \mathcal{L} .

4 HEURÍSTICAS

Nesta seção, apresentamos um conjunto de heurísticas construtivas para o problema, bem como heurísticas baseadas nos modelos de *Set Partitioning*.

As heurísticas construtivas implementadas tiveram como base heurísticas propostas em problemas de empacotamento, como apresentadas em (RIECK, 2021) para problema da mochila binária, em (HAOUARI; SERAIRI, 2009) para problema da mochila binária com mochilas de tamanhos variáveis e em (CHRISTENSEN et al., 2017) para problema de empacotamento multidimensional.

4.1 Heurística Aleatória

Na heurística aleatória (HA), a lista de pedidos é percorrida. O pedido poderá dar início a um novo lote ou poderá haver uma tentativa de inseri-lo em um lote já existente. Esta decisão é tomada aleatoriamente. Neste último caso, se não for possível inseri-lo em nenhum lote já existente, em razão da capacidade, um novo lote será criado. Algoritmo exemplificado na Figura 4.1.

Figura 4.1 – Heurísticas de solução - Aleatória

```

for (pedido in pedidos) {
  do {
    chance = random(0,1);
    if(chance == 0) {
      lote = lotes.loteAleatorio();
      lote.tentarInserirPedido(pedido);
    }
    else if(chance == 1) {
      novoLote.inserirPedido(pedido);
      lotes.adicionarLote(novoLote);
    }
  } while (pedidoNaoInserido);
}

```

Fonte: Elaborada pelo autor (2022)

4.2 Heurística Gulosa

Na heurística gulosa (HG), a lista de pedidos é percorrida. Haverá uma tentativa de inserir o pedido corrente em um lote já existente. Caso seja possível, a inserção ocorrerá no primeiro lote encontrado capaz de suportar o pedido. Caso contrário, um novo lote será criado. Algoritmo exemplificado na Figura 4.2.

Figura 4.2 – Heurísticas de solução - Gulosa

```

for(pedido in pedidos) {
  for(lote in lotes) {
    if(pedido cabe em lote) {
      lote.inserirPedido(pedido);
      break;
    }
  }
  if(pedidoNaoInserido) {
    novoLote.inserirPedido(pedido);
    lotes.adicionarLote(novoLote);
  }
}

```

Fonte: Elaborada pelo autor (2022)

4.3 Heurística Gulosa Ordenada

A heurística gulosa ordenada (HGO) é similar à heurística gulosa acima, porém os pedidos são ordenados em ordem não crescente de sua ocupação no equipamento de coleta, isto é, os produtos que ocupam mais são considerados primeiro. Algoritmo exemplificado na Figura 4.3.

Figura 4.3 – Heurísticas de solução - Gulosa ordenada

```

pedidos = pedidos.ordenarPorTamanhoMajoresPrimeiro();
for(pedido in pedidos) {
  for(lote in lotes) {
    if(pedido cabe em lote) {
      lote.inserirPedido(pedido);
      break;
    }
  }
  if(pedidoNaoInserido) {
    novoLote.inserirPedido(pedido);
    lotes.adicionarLote(novoLote);
  }
}

```

Fonte: Elaborada pelo autor (2022)

4.4 Heurística Gulosa por Similaridade de Rota

Na heurística gulosa por similaridade de rota (HGS), os pedidos também são considerados em ordem não crescente de sua ocupação. Para cada pedido, é verificada sua “similaridade de rota” com os lotes já existentes.

A similaridade é definida por $100 \times CE/CT$, onde CE é a quantidade de corredores necessários para pedido e que já são necessários naquele lote, e CT é a quantidade total de corredores necessários para o pedido.

Se a maior similaridade for maior que um parâmetro, então o pedido é inserido no lote correspondente, caso caiba. Caso contrário, o pedido é inserido de acordo com a heurística gulosa. Neste trabalho foi usado o valor 50 para o parâmetro, sendo esse um valor definido após testes com valores 30, 40, 50, 60 e 70, onde o 50 obteve melhores resultados. Algoritmo exemplificado na Figura 4.4.

Figura 4.4 – Heurísticas de solução - Gulosa por similaridade de rota

```

for(pedido in pedidos) {
  for(lote in lotes) {
    lote.similar = calcularSimilar(pedido, lote);
  }
  if(maiorSimilaridade(lotes) > minimoSimilaridade) {
    loteMaiorSimilaridade.inserirPedido(pedido);
  }
  else {
    insercaoGulosa();
  }
}

```

Fonte: Elaborada pelo autor (2022)

4.5 Gulosa por diferença de rota

Na heurística gulosa por diferença de rota (HGD), os pedidos também são considerados em ordem não crescente de sua ocupação. Para cada pedido, é considerada sua “diferença de rota” com os lotes já existentes.

A diferença é definida pela quantidade de novos corredores que serão adicionados ao lote, caso o pedido seja incluído nele.

Se a menor diferença for menor que um parâmetro, então o pedido é inserido no lote correspondente, caso caiba. Caso contrário, o pedido é inserido de acordo com a heurística gulosa. Neste trabalho foi usado o valor 3 para o parâmetro, sendo esse um valor definido após testes com valores 1, 2, 3, 4 e 5, onde o 3 obteve melhores resultados. Algoritmo exemplificado na Figura 4.5.

4.6 Heurísticas Baseadas nos Modelos de *Set Partitioning*

As heurísticas baseadas nos modelos de *Set Partitioning* (HBSPs) consistem em resolver os respectivos modelos, porém com conjuntos restritos \mathcal{L} , cujos lotes são gerados previamente, por meio da heurística aleatória, descrita acima.

Figura 4.5 – Heurísticas de solução - Gulosa por diferença de rota

```
for(pedido in pedidos) {  
  for(lote in lotes) {  
    lote.diferencas = calcularDiferenca(pedido, lote);  
  }  
  if(menorDiferenca(lotes) > diferencaMaxima) {  
    loteMenorDiferencas.inserirPedido(pedido);  
  }  
  else {  
    insercaoGulosa();  
  }  
}
```

Fonte: Elaborada pelo autor (2022)

A heurística aleatória irá ordenar aleatoriamente a lista de pedidos e percorrê-la, inserindo-os aleatoriamente nos lotes gerados até o momento. Para cada pedido, os lotes serão percorridos em ordem de criação. Com uma probabilidade de 30% será realizada uma tentativa de inserção no lote atualmente selecionado. Caso não seja possível inserir o pedido, em razão da capacidade do lote, o processo continuará a partir do lote seguinte. Não sendo possível inserir o pedido em nenhum dos lotes, um novo lote será criado. O parâmetro 30% foi definido após testes com valores 20%, 30%, 40% e 50%, onde o 30% obteve melhores resultados.

5 EXPERIMENTOS COMPUTACIONAIS

Nesta seção, apresentamos resultados computacionais para as heurísticas apresentadas neste trabalho e também para um algoritmo exato obtido ao se introduzirem os modelos compactos da Seção 3.1, aplicando a modelagem dos tempos até o último corredor devido a sua velocidade de execução, em um resolvidor comercial. Os algoritmos foram todos implementados na linguagem de programação Java. Foi utilizado o resolvidor comercial Gurobi, versão 9.1.2, com configuração padrão. Todos os testes foram realizados em um Intel Core I7 de décima geração, com 4 núcleos e 8 processadores lógicos, sistema operacional Windows 10.

5.1 Instâncias

Para definir as instâncias do problema, tentamos contatos para obter dados reais, porém não foi possível obter todas as informações necessárias, como a organização do armazém ou um histórico de pedidos. Sendo assim, a partir da pesquisa realizada no referencial teórico, definimos três tipos de instâncias. Os equipamentos de coletas das instâncias são idênticos para todas as zonas, com um tamanho de 30 unidades. Os produtos gerados tem tamanhos aleatórios, variando de 1 até o tamanho do equipamento de coleta.

Nas instâncias do tipo 1 os pedidos possuem objetos sem relação entre si, espalhados aleatoriamente pelo armazém. Nas instâncias do tipo 2, pedidos tendem a ter produtos mais próximos da primeira zona, ou seja, a probabilidade da zona 1 é maior que a da 2, a probabilidade da zona 2 é maior que a da 3, e assim por diante. Instâncias do tipo 3 são compostas de pedidos cujos objetos tendem a estar na mesma zona. Em todas as instâncias o armazém possui 5 zonas com 10 corredores verticais em cada. É considerado que cada corredor vertical possui 20 metros e a distância entre cada um deles é de 3 metros. Produtos podem estar localizados em qualquer ponto do corredor vertical.

Podemos interpretar instâncias do tipo 1 como instâncias cujos armazéns estão desorganizados. Instâncias do tipo 3 possuem armazéns bem organizados, de acordo com os padrões de compra dos clientes. Já instâncias do tipo 2 possuem armazéns organizados para uma situação diferente da qual estão sendo utilizados.

5.2 Resultados

Os resultados são apresentados no Quadro 5.1. Nas quatro primeiras colunas são apresentadas informações sobre as instâncias: tipo (T); número de produtos (N); número total de objetos, na média (QP); estratégia de coleta empregada, progressiva ou sincronizada (P/S). As duas colunas seguintes apre-

sentam o menor limite superior (UB) e o maior limite inferior (LB) encontrados ao longo da exploração da árvore de branch-and-bound, obtida através da execução do resolvidor comercial. Os resultados das heurísticas são apresentados nas quatro colunas seguintes: aleatória (HA), gulosa (HG), gulosa ordenada (HGO), gulosa por similaridade de rota (HGS), gulosa por diferença de rota (HGD), e baseada no modelo de *Set Partitioning* (HBSP). Cada linha do Quadro corresponde à média dos resultados em cinco instâncias. Os modelos compactos e as heurísticas baseadas nos modelos de *Set Partitioning* foram executados com um limite de tempo de uma hora. Todas as heurísticas construtivas foram executadas em menos de um segundo.

Cada par de linhas do Quadro corresponde a um mesmo conjunto de instâncias, sendo que a primeira linha reporta resultados para a estratégia progressiva e a segunda para a estratégia sincronizada. Nota-se que a estratégia sincronizada produz soluções de custo significativamente menor em instâncias do tipo 1 e 2, 28% e 22%, respectivamente, considerando instâncias com 200 pedidos e o melhor resultado para cada estratégia. Como o limitante de combinações é a capacidade do equipamento de coleta, a estratégia progressiva tem uma capacidade de combinação de lotes muito menor, pois o limite é definido para todo o armazém, enquanto na sincronizada o limite é por zona, sendo possível realizar melhores combinações dos pedidos em lotes. Dessa forma, não havendo um motivo maior para o uso da estratégia progressiva, nos parece melhor empregar a estratégia sincronizada.

Dentre as heurísticas construtivas, dos 24 grupos de instâncias, HGD foi melhor em 18, HG foi melhor em 5, HGS em 1, onde apenas as maiores instâncias do tipo 3 tiveram um resultado um pouco pior na abordagem sincronizada. Para a estratégia progressiva, o desempenho das heurísticas construtivas não é significativamente melhor que o da heurística aleatória. Já no caso da estratégia sincronizada, a diferença foi maior que 80% ao comparar o melhor resultado com as heurísticas HA e HGS. Na zona sincronizada, lotes que são mais balanceados entre as zonas irão obter um melhor resultado, evitando tempo ocioso dos funcionários. Como a HGS irá criar lotes que tendencialmente sobrecarregam alguma zona, terá um desbalanceamento, gerando um pior desempenho dessa heurística, sendo inclusive pior que a HA nos testes realizados.

O HBSP teve um desempenho melhor que todas as heurísticas. Considerando problemas com 200 pedidos, para a estratégia progressiva, seu resultado, quando comparado ao melhor resultado de alguma heurística construtiva, foi em média 17%, 3% e 15% melhor, respectivamente, para instâncias do tipo 1, 2, e 3. Para a estratégia sincronizada, os respectivos números são 13%, 8% e 10%.

Quando comparamos HBSP com os modelos compactos, no caso da estratégia progressiva, vemos que HBSP foi capaz de fornecer limites superiores melhores, 10% para as maiores instâncias do tipo 1, e menos de 1% para as instâncias correspondentes dos tipos 2 e 3. Para a estratégia sincronizada, os

respectivos valores são 12%, 0,5% e -7%. Ou seja, o desempenho de HBSP foi pior que o do modelo compacto para instâncias do tipo 3. No geral, podemos concluir que HBSP apresenta os melhores limites superiores dentro do tempo de uma hora.

Os resultados com instâncias menores, e contendo mais detalhes, além da evolução das abordagens para solução, estão no Apêndice A.

No geral, podemos concluir que o *Set Partitioning* apresenta os melhores limites superiores dentro do tempo de uma hora. Apesar disso, é possível que esta abordagem apresentasse resultados ainda melhores caso fosse inicializada com um conjunto restrito \mathcal{L} de melhor qualidade, tal como o conjunto restrito obtido ao fim da resolução da relaxação linear do modelo baseado em *Set Partitioning* por meio de geração de colunas.

Quadro 5.1 – Resultados dos experimentos computacionais

T	Instância		Modelo Compacto						Heurísticas					
	N	QP	P/S	LB	UB	HA	HG	HGO	HGS	HGD	HBSP			
1	50	106,6	P	9456,2	9753,6	13512,6	13454	13780,2	13856,6	13780,2	9753,6			
			S	5104	7075,2	16936,6	9411	9607,8	16551	9616,6	7264			
	100	210,4	P	13507	18736,8	24273,2	23977,2	24410,8	24404,4	24287,8	18717,4			
			S	5865,8	13240,8	36301,6	17063,2	17193,4	29724	16412,6	13507,6			
	150	319,2	P	20479,2	27605	34920,4	34603,2	34211,4	33854,8	33951,8	27568,8			
			S	7824,4	22048	52356,6	24789,4	24993,2	39941,8	24071,6	20599,8			
	200	438,8	P	14389,8	41804,8	46980,4	46559	45651	45298,8	45212,6	37580,2			
			S	9869,8	33462,2	70950,2	34053,4	33859	51188	32766,4	29193,2			
2	50	108,4	P	18275	18313,6	19556,4	19189,2	19559,6	19651,8	19328	18313,6			
			S	11798,8	13553,2	21174,2	16078,6	16235,4	19114,6	15571	13649			
	100	212,8	P	34441,6	35138	38336,2	37013,2	37291	37154	37010,6	35099,2			
			S	20327,2	25871	41765,6	30644,6	30849,2	36289,6	30090,2	27214,8			
	150	321	P	42738,4	51928	55579,4	54474,2	53650,2	53919,4	53352,4	51863,4			
			S	28895,2	38265,2	60444,6	46109,4	45344	50557,2	43816	37909			
	200	429,6	P	55951,6	69965,2	74951,4	73186,4	72922,8	72107,2	71481,6	69922,6			
			S	25990,4	57572,2	84204,2	63225,4	62352,2	67586,4	59233,4	57249,8			
3	50	105,2	P	7540,8	8207,4	11383,4	11413,4	11156,6	11164,2	10962,2	8207,4			
			S	7166,2	8329,8	18316,2	10156,2	10135	26743	10018,6	9120,4			
	100	214,2	P	12419	14164	18951,6	18939,2	20049,2	19572	19602,6	14148,2			
			S	11515,8	14363	40107,2	17940,4	17984,4	51973,2	17721,8	16080,6			
	150	323,6	P	19059	23493,4	30181	29413	28369,8	27926,2	27685,4	23478			
			S	18125,2	22989	60472,6	28717,4	28833,4	76648,4	28406	26346,2			
	200	430,2	P	14856,6	30809,6	39267,6	39410	37134,8	36127,2	36098,2	30658,2			
			S	18465,4	30417,4	79832,6	37416,4	36361,8	99177,8	36144,8	32602,6			

Fonte: Elaborado pelo autor (2022)

6 CONCLUSÃO

Neste trabalho, foi abordado o problema de coleta de pedidos em armazéns divididos em zonas. Mais especificamente, como organizar os pedidos em lotes de forma que o tempo total de coleta seja mínimo. Foram definidas instâncias e propostas soluções, como modelos compactos, baseadas em *Set Partitioning* e heurísticas, executadas para os casos de zonas progressivas e sincronizadas.

Foram realizados experimentos computacionais com três tipos de instâncias, representando diferentes formas de organização dos pedidos nos armazéns. A análise de resultados apresenta que os melhores limites superiores foram obtidos pela solução em *Set Partitioning*.

Em trabalhos futuros, é possível aprofundar na investigação de estratégias baseadas em *Set Partitioning*, tais como algoritmos de geração de colunas, para avaliar os limites de relaxação linear dos modelos e heurísticas do tipo *diving* (SADYKOV et al., 2019).

Com o aumento das instâncias e limitação de tempo de execução, alternativas diferentes aos modelos compactos têm espaço de estudo, como estratégias de busca local. Além disso, implementação de novas heurísticas, principalmente que consigam gerar lotes mais balanceados, ou heurísticas que considerem adaptações específicas de acordo com o tipo de zona, visto que zonas progressivas são impactadas pela ordem que os lotes forem coletados, também são possibilidades de trabalhos futuros.

Os resultados obtidos são considerando a heurística de roteamento S-shape, sendo possível melhorar o tempo de coleta caso otimize o roteamento do agrupamento definido.

Os modelos utilizados neste trabalho consideram a quantidade de lotes possíveis como sendo a quantidade total de pedidos. Essa definição pode gerar muitas variáveis no modelo, onde muitas delas não são utilizadas na solução final, sendo possível a geração de diversas soluções simétricas. Portanto, como trabalho futuro, também é possível melhorar a definição dos limites das variáveis no modelo, buscando uma melhor velocidade na execução deles.

A estrutura do armazém e das zonas é muito impactante para a definição dos melhores agrupamentos, alterar a abordagem de criação de zonas, sendo dinâmicas ou com quantidades diferentes de corredores em casa zona, também são adaptações que podem gerar melhorias.

REFERÊNCIAS

- AMARAL, A. R. A parallel ordering problem in facilities layout. **Computers & Operations Research**, Elsevier BV, v. 40, n. 12, p. 2930–2939, dec 2013.
- BAKER, P.; HALIM, Z. An exploration of warehouse automation implementations: cost, service and flexibility issues. **Supply Chain Management: An International Journal**, Emerald, v. 12, n. 2, p. 129–138, mar 2007.
- BOYSEN, N.; KOSTER, R. de; WEIDINGER, F. Warehousing in the e-commerce era: A survey. **European Journal of Operational Research**, Elsevier BV, v. 277, n. 2, p. 396–411, sep 2019.
- CALZAVARA, M. et al. Analysis of economic and ergonomic performance measures of different rack layouts in an order picking warehouse. **Computers & Industrial Engineering**, Elsevier BV, v. 111, p. 527–536, sep 2017.
- CHEN, C.-M. et al. A flexible evaluative framework for order picking systems. **Production and Operations Management**, Wiley, v. 19, n. 1, p. 70–82, jan 2010.
- CHRISTENSEN, H. I. et al. Approximation and online algorithms for multidimensional bin packing: A survey. **Computer Science Review**, Elsevier BV, v. 24, p. 63–79, 2017.
- DALLARI, F.; MARCHET, G.; MELACINI, M. Design of order picking system. **The International Journal of Advanced Manufacturing Technology**, Springer Science and Business Media LLC, v. 42, n. 1-2, p. 1–12, jun 2008.
- DIJKSTRA, A. S.; ROODBERGEN, K. J. Exact route-length formulas and a storage location assignment heuristic for picker-to-parts warehouses. **Transportation Research Part E: Logistics and Transportation Review**, Elsevier BV, v. 102, p. 38–59, jun 2017.
- EKREN, B.; SARI, Z.; LERHER, T. Warehouse design under class-based storage policy of shuttle-based storage and retrieval system. **IFAC-PapersOnLine**, Elsevier BV, v. 48, n. 3, p. 1152–1154, 2015.
- GILS, T. van et al. Designing efficient order picking systems by combining planning problems: State-of-the-art classification and review. **European Journal of Operational Research**, Elsevier BV, v. 267, n. 1, p. 1–15, may 2018.
- HAOUARI, M.; SERAIRI, M. Heuristics for the variable sized bin-packing problem. **Computers & Operations Research**, Elsevier BV, v. 36, n. 10, p. 2877–2884, 2009.
- HENN, S.; WÄSCHER, G. Tabu search heuristics for the order batching problem in manual order picking systems. **European Journal of Operational Research**, Elsevier BV, v. 222, n. 3, p. 484–494, nov 2012.
- HO, Y.-C.; LIN, J.-W. Improving order-picking performance by converting a sequential zone-picking line into a zone-picking network. **Computers & Industrial Engineering**, Elsevier BV, v. 113, p. 241–255, nov 2017.
- HSIEH, L.-F.; HUANG, Y.-C. New batch construction heuristics to optimise the performance of order picking systems. **International Journal of Production Economics**, Elsevier BV, v. 131, n. 2, p. 618–630, jun 2011.
- HUANG, M. et al. Mixed model assembly line scheduling approach to order picking problem in online supermarkets. **Sustainability**, MDPI AG, v. 10, n. 11, p. 3931, 2018.

- IVANOV, A. E. The internet's impact on integrated marketing communication. **Procedia Economics and Finance**, Elsevier BV, v. 3, p. 536–542, 2012.
- JANE, C.-C.; LAIH, Y.-W. A clustering algorithm for item assignment in a synchronized zone order picking system. **European Journal of Operational Research**, Elsevier BV, v. 166, n. 2, p. 489–496, oct 2005.
- JEWKES, E.; LEE, C.; VICKSON, R. Product location, allocation and server home base location for an order picking line with multiple servers. **Computers & Operations Research**, Elsevier BV, v. 31, n. 4, p. 623–636, apr 2004.
- KOSTER, R. B. de; LE-DUC, T.; ZAERPOUR, N. Determining the number of zones in a pick-and-sort order picking system. **International Journal of Production Research**, Informa UK Limited, v. 50, n. 3, p. 757–771, feb 2012.
- LI, X. Valuing lead-time and its variance in batch-ordering inventory policies. **International Journal of Production Economics**, Elsevier BV, v. 228, p. 107731, oct 2020.
- MARCHET, G.; MELACINI, M.; PEROTTI, S. Investigating order picking system adoption: a case-study-based approach. **International Journal of Logistics Research and Applications**, Informa UK Limited, v. 18, n. 1, p. 82–98, aug 2014.
- MASAE, M.; GLOCK, C. H.; GROSSE, E. H. Order picker routing in warehouses: A systematic literature review. **International Journal of Production Economics**, Elsevier BV, v. 224, p. 107564, jun 2020.
- NIEUWENHUYSE, I. V.; KOSTER, R. B. de. Evaluating order throughput time in 2-block warehouses with time window batching. **International Journal of Production Economics**, Elsevier BV, v. 121, n. 2, p. 654–664, oct 2009.
- OZDEN, S. G.; SMITH, A. E.; GUE, K. R. A computational software system to design order picking warehouses. **Computers & Operations Research**, Elsevier BV, v. 132, p. 105311, 2021.
- PAN, J. C.-H.; SHIH, P.-H.; WU, M.-H. Order batching in a pick-and-pass warehousing system with group genetic algorithm. **Omega**, Elsevier BV, v. 57, p. 238–248, dec 2015.
- PAN, J. C.-H. et al. A storage assignment heuristic method based on genetic algorithm for a pick-and-pass warehousing system. **Computers & Industrial Engineering**, Elsevier BV, v. 81, p. 1–13, mar 2015.
- QUADER, S.; CASTILLO-VILLAR, K. K. Design of an enhanced multi-aisle order-picking system considering storage assignments and routing heuristics. **Robotics and Computer-Integrated Manufacturing**, Elsevier BV, v. 50, p. 13–29, apr 2018.
- RAKESH, V.; ADIL, G. K. Layout optimization of a three dimensional order picking warehouse. **IFAC-PapersOnLine**, Elsevier BV, v. 48, n. 3, p. 1155–1160, 2015.
- RIECK, B. A. Basic analysis of bin-packing heuristics. **ArXiv**, abs/2104.12235, 2021.
- SADYKOV, R. et al. Primal heuristics for branch and price: The assets of diving methods. **INFORMS Journal on Computing**, INFORMS, v. 31, n. 2, p. 251–267, 2019.
- SILVA, A. et al. Integrating storage location and order picking problems in warehouse planning. **Transportation Research Part E: Logistics and Transportation Review**, Elsevier BV, v. 140, p. 102003, aug 2020.

VALLE, C. A.; BEASLEY, J. E. Order batching using an approximation for the distance travelled by pickers. **European Journal of Operational Research**, Elsevier BV, v. 284, n. 2, p. 460–484, 2020.

VALLE, C. A.; BEASLEY, J. E.; CUNHA, A. S. da. Optimally solving the joint order batching and picker routing problem. **European Journal of Operational Research**, Elsevier BV, v. 262, n. 3, p. 817–834, nov 2017.

YU, M.; KOSTER, R. B. de. The impact of order batching and picking area zoning on order picking system performance. **European Journal of Operational Research**, Elsevier BV, v. 198, n. 2, p. 480–490, oct 2009.

A RESULTADOS DETALHADOS

A.1 Primeira etapa de testes

Para a primeira etapa de testes, todas as instâncias utilizadas contém 10 pedidos, para que fosse possível validar o modelo com a primeira modelagem dos tempos de processamento dos lotes, o processamento de trajeto, apresentada na Seção 3.1.3.1. A execução do modelo consegue encontrar a solução ótima na maioria dos casos, em um tempo bem menor do que uma hora. Como a zona progressiva permite menos combinações, devido ao tamanho do equipamento de coleta, ela é executada com maior velocidade.

Alguns casos da zona sincronizada não conseguiram encontrar a solução ótima em uma hora de processamento, principalmente para instâncias de tipo 2, pois todos os pedidos estão tendencialmente próximos, aumentando as combinações possíveis. Para a maioria das instâncias testadas, o modelo obteve um limite inferior melhor na abordagem sincronizada.

As heurísticas de solução foram executadas em menos de um segundo para qualquer instância testada. Os resultados para zona progressiva são bem piores em relação aos do modelo, pois a ordem de coleta dos lotes influencia no resultado. Para zona sincronizada os resultados foram muito mais próximos, e em alguns casos iguais, aos obtidos no modelo.

Devido ao baixo número de pedidos, para zona progressiva, as heurísticas que consideram similaridade e diferença de rotas geraram praticamente os mesmos resultados do que a gulosa ordenada, enquanto a heurística aleatória obteve resultados parecidos com a gulosa. Isso ocorre pelo baixo número de combinações possíveis, provavelmente as instâncias possuem pedidos muito grandes. Sendo assim, é difícil definir a melhor heurística para zona progressiva com dez pedidos.

Na zona sincronizada, devido ao maior número de combinações, temos resultados mais próximos aos da segunda etapa. A heurística com similaridade de rotas é muito pior para tal tipo de zona, enquanto as demais heurísticas gulosas geram as melhores soluções.

Com base nos resultados, percebemos que o tempo de processamento não estava baixo, portanto replicamos a primeira etapa de testes para as heurísticas propostas, para a segunda modelagem dos tempos de processamento dos lotes, apresentada na Seção 3.1.3.2, e para o *Set Partitioning*, com resultados promissores, sendo necessária uma validação com instâncias maiores. Definimos também a segunda etapa de testes, com instâncias onde a quantidade de pedidos varia entre 10 e 50.

A.2 Segunda etapa de testes

Para a segunda etapa de testes, todas as instâncias utilizadas contém entre 10 e 50 pedidos, para que fosse possível validar os dois modelos de cálculo de tempo e as heurísticas. A execução do modelo, durante uma hora, deixa um gap considerável, sendo possível uma grande melhoria de solução caso seja executado por mais tempo. Porém, para as instâncias testadas, o modelo obteve resultado melhor do que as heurísticas de solução propostas.

O modelo necessita de mais tempo para zona sincronizada, visto que é possível atribuir mais pedidos a um lote. Nas instâncias testadas, mesmo com lotes pequenos, em alguns casos não foi possível obter a solução ótima após uma hora, enquanto para zona progressiva foi possível percorrer a maior parte das soluções.

Para a maioria das instâncias testadas, o modelo obteve um limite inferior melhor na abordagem sincronizada. É importante notar que, a abordagem sincronizada permite uma maior quantidade de lotes gerados, devido à capacidade total do equipamento de coleta, enquanto a abordagem progressiva precisa sempre validar o tempo de cada zona, e a ordem dos lotes gerados é importante para o resultado.

Para a abordagem de zona progressiva, a ordem dos lotes impacta no resultado. Sendo assim, melhorias após a definição dos lotes, alterando a ordem que estão planejados para coleta, poderão gerar melhores resultados.

A heurística que considera similaridade de rotas, quando aplicada para zona sincronizada, apresenta resultados muito piores, visto que algumas zonas ficarão sobrecarregadas, enquanto outras ficam sem trabalho. Uma heurística que gera lotes balanceados entre as zonas pode apresentar melhores resultados neste caso.

As instâncias do tipo 2, com uma maior chance de todos os pedidos concentrarem produtos na mesma zona, geram resultados piores, devido a sobrecarga das zonas. Independente de qual a abordagem, é importante que todas as zonas tenham demandas, para que seja reduzido o tempo de coleta. Nestas situações, gerar zonas com quantidade diferente de corredores poderá gerar melhores resultados.

Para os demais tipos de instância, os resultados obtidos quanto ao tempo de coleta é melhor. Para instâncias do tipo 1, as aleatórias, é possível gerar lotes com produtos nas diversas zonas. Para instâncias do tipo 3, onde um pedido contém produtos tendencialmente na mesma zona, porém cada pedido pode ter foco em uma zona diferente, geram situações que, para zona sincronizada, é fácil a combinação de pedidos em lotes, enquanto que, para zona progressiva, é possível sincronizar os lotes e balancear melhor o esforço por zona.

A segunda modelagem de cálculo de tempo apresenta resultados bem melhores do que a primeira modelagem, porém, ao aumentar o tamanho das instâncias, até a segunda modelagem começa a apresentar dificuldades para encontrar a solução ótima no tempo limite definido. Sendo assim, replicamos esta etapa de testes com o *Set Partitioning*, para conseguir resultados melhores que as heurísticas, e soluções melhores que o modelo no mesmo tempo limite de processamento. Definimos a terceira etapa de testes com instâncias maiores, variando entre 50 e 200 pedidos.

Resultados da terceira etapa dos testes são apresentados no Capítulo 5.

A.3 Resultados detalhados

Nomenclaturas utilizadas nas representações a seguir:

- a) Tipo, denota o tipo da instância, entre os três implementados;
- b) N , denota o número de pedidos;
- c) QP , denota a quantidade total de produtos de uma instância, sendo o somatório dos produtos de todos os pedidos;
- d) HA , Heurística aleatória (4.1);
- e) HG , Heurística gulosa (4.2);
- f) HGO , Heurística gulosa ordenada (4.3);
- g) HGS , Heurística gulosa com similaridade (4.4);
- h) HGD , Heurística gulosa com diferença (4.5);
- i) RM_1 , cálculo de tempo com processamento de trajeto (3.1.3.1);
- j) RM_2 , cálculo de tempo com distância até o último corredor (3.1.3.2);
- k) $HBSP$, melhor resultado do *Set Partitioning* (4.6);
- l) RM , denota a melhor solução do modelo;
- m) TM , denota o tempo total de execução, em segundos, do modelo;
- n) TMM , denota o tempo de execução, em segundos, do modelo, até encontrar a melhor solução obtida;
- o) LI , denota o melhor limite inferior ao final da execução;
- p) GAP , denota o intervalo entre a solução encontrada e o limite inferior ao final da execução.
($100(RM - LI)/RM$);
- q) LR , denota o limite da relaxação linear;
- r) $RMAP$, denota a pior solução do modelo com a heurística aleatória;
- s) $RMAM$, denota a melhor solução do modelo com a heurística aleatória;
- t) $RMSP$, denota a pior solução do modelo com a heurística por similaridade de rota;

- u) *RMSM*, denota a melhor solução do modelo com a heurística por similaridade de rota;
- v) *TMC*, denota o tempo médio de execução, em segundos, do modelo para combinação de lotes;

Quadro A.1 – Resultados da primeira modelagem de cálculo de tempo, na primeira etapa, para zona progressiva

Tipo	<i>N</i>	<i>QP</i>	<i>RM</i>	<i>TM</i>	<i>TMM</i>	<i>LI</i>	<i>GAP</i>	<i>LR</i>
1	10	21	6336	18	3	6336	0%	132
1	10	21	7754	28	10	7754	0%	148
1	10	21	7245	52	26	7245	0%	196
1	10	19	7990	54	14	7990	0%	193
1	10	18	8773	5	3	8773	0%	126
2	10	24	10110	167	117	10110	0%	188
2	10	26	9275	60	34	9275	0%	287
2	10	18	8223	57	11	8223	0%	163
2	10	26	8421	69	46	8421	0%	195
2	10	30	13120	86	4	13120	0%	226
3	10	25	9441	218	99	9441	0%	222
3	10	21	12926	412	1	12926	0%	253
3	10	22	6299	60	17	6299	0%	206
3	10	19	7186	23	5	7186	0%	150
3	10	20	9415	118	6	9415	0%	147

Fonte: Elaborado pelo autor (2022)

Quadro A.2 – Resultados da primeira modelagem de cálculo de tempo, na segunda etapa, para zona progressiva

Tipo	<i>N</i>	<i>QP</i>	<i>RM</i>	<i>TM</i>	<i>TMM</i>	<i>LI</i>	<i>GAP</i>	<i>LR</i>
1	10	20	858	27	1	858	0,00%	59
1	20	37	1554	3600	17	1068	21,27%	153
1	30	57	2140	3600	142	1185	44,61%	168
1	40	89	3626	3600	826	1460	59,72%	308
1	50	103	3532	3600	2954	1288	63,52%	261
2	10	23	1452	575	1	1452	0,00%	192
2	20	40	17000	3600	710	8371	50,76%	358
2	30	61	4281	3600	2627	2825	34,00%	383
2	40	87	4848	3600	729	1688	65,16%	416
2	50	113	5400	3600	1257	2277	57,82%	540
3	10	29	818	20	5	818	0,00%	80
3	20	43	1253	3600	2485	876	30,08%	120
3	30	62	1699	3600	12	963	43,28%	199
3	40	86	2196	3600	3342	1583	27,71%	310
3	50	100	2983	3600	349	1598	46,41%	383

Fonte: Elaborado pelo autor (2022)

Quadro A.3 – Resultados da segunda modelagem de cálculo de tempo, na primeira etapa, para zona progressiva

Tipo	<i>N</i>	<i>QP</i>	<i>RM</i>	<i>TM</i>	<i>TMM</i>	<i>LI</i>	<i>GAP</i>	<i>LR</i>
1	10	21	6336	< 1	< 1	6336	0%	3716
1	10	21	7754	< 1	< 1	7754	0%	3187
1	10	21	7245	< 1	< 1	7245	0%	2354
1	10	19	7990	< 1	< 1	7990	0%	2646
1	10	18	8773	< 1	< 1	8773	0%	3008
2	10	24	10110	< 1	< 1	10110	0%	2889
2	10	26	9275	< 1	< 1	9275	0%	2791
2	10	18	8223	< 1	< 1	8223	0%	2615
2	10	26	8421	< 1	< 1	8421	0%	2694
2	10	30	13120	< 1	< 1	13120	0%	3351
3	10	25	9441	< 1	< 1	9441	0%	2553
3	10	21	12926	< 1	< 1	12926	0%	2797
3	10	22	6299	< 1	< 1	6299	0%	2350
3	10	19	7186	< 1	< 1	7186	0%	2887
3	10	20	9415	1	< 1	9415	0%	2630

Fonte: Elaborado pelo autor (2022)

Quadro A.4 – Resultados da segunda modelagem de cálculo de tempo, na segunda etapa, para zona progressiva

Tipo	<i>N</i>	<i>QP</i>	<i>RM</i>	<i>TM</i>	<i>TMM</i>	<i>LI</i>	<i>GAP</i>	<i>LR</i>
1	10	20	858	< 1	< 1	858	0,00%	342
1	20	37	1554	47	37	1554	0,00%	406
1	30	57	2140	12	2	2140	0,00%	420
1	40	89	3626	3600	440	3622	0,11%	561
1	50	103	3532	3600	2368	3528	0,11%	514
2	10	23	1452	1	< 1	1452	0,00%	444
2	20	40	17000	3	< 1	17000	0,00%	2532
2	30	61	4281	64	38	4821	0,00%	636
2	40	87	4848	3600	133	4846	0,04%	668
2	50	113	5400	3600	14	5396	0,07%	792
3	10	29	818	< 1	< 1	818	0,00%	345
3	20	43	1253	6	1	1253	0,00%	372
3	30	62	1699	104	17	1699	0,00%	451
3	40	86	2196	490	8	2196	0,00%	562
3	50	100	2983	530	330	2983	0,00%	635

Fonte: Elaborado pelo autor (2022)

Quadro A.5 – Resultados do *Set Partitioning*, na primeira etapa, para zona progressiva

Tipo	<i>N</i>	<i>QP</i>	<i>RMAP</i>	<i>RMAM</i>	<i>RMSP</i>	<i>RMSM</i>	<i>TMC</i>
1	10	21	6336	6336	6336	6336	< 1
1	10	21	7754	7754	7754	7754	< 1
1	10	21	7245	7245	7245	7245	< 1
1	10	19	7990	7990	7990	7990	< 1
1	10	18	8773	8773	8773	8773	< 1
2	10	24	10110	10110	10110	10110	< 1
2	10	26	9275	9275	9275	9275	< 1
2	10	18	8223	8223	8223	8223	< 1
2	10	26	8421	8421	8421	8421	< 1
2	10	30	13120	13120	13120	13120	< 1
3	10	25	9441	9441	9441	9441	< 1
3	10	21	12926	12926	12926	12926	< 1
3	10	22	6299	6299	6299	6299	< 1
3	10	19	7186	7186	7186	7186	< 1
3	10	20	9415	9415	9415	9415	< 1

Fonte: Elaborado pelo autor (2022)

Quadro A.6 – Resultados do *Set Partitioning*, na segunda etapa, para zona progressiva

Tipo	<i>N</i>	<i>QP</i>	<i>RMAP</i>	<i>RMAM</i>	<i>RMSP</i>	<i>RMSM</i>	<i>TMC</i>
1	10	20	858	858	858	858	< 1
1	20	37	1554	1554	1554	1554	< 1
1	30	57	2140	2140	2140	2140	< 1
1	40	89	3626	3626	3626	3626	< 1
1	50	103	3532	3532	3532	3532	< 1
2	10	23	1452	1452	1452	1452	< 1
2	20	40	17000	17000	17000	17000	< 1
2	30	61	4281	4281	4281	4281	< 1
2	40	87	4848	4848	4848	4848	< 1
2	50	113	5400	5400	5400	5400	< 1
3	10	29	818	818	818	818	< 1
3	20	43	1253	1253	1253	1253	< 1
3	30	62	1699	1699	1699	1699	< 1
3	40	86	2196	2196	2196	2196	< 1
3	50	100	2983	2983	2983	2983	< 1

Fonte: Elaborado pelo autor (2022)

Quadro A.7 – Resultados do *Set Partitioning*, na terceira etapa com instâncias de tamanho 50, para zona progressiva

Tipo	<i>N</i>	<i>QP</i>	<i>RM</i>	<i>RMAP</i>	<i>RMAM</i>	<i>TMC</i>
1	50	90	9852	9852	9852	< 1
1	50	102	9181	9181	9181	< 1
1	50	111	8866	8866	8866	< 1
1	50	117	9197	9197	9197	< 1
1	50	113	11672	11672	11672	< 1
2	50	101	15803	15812	15803	< 1
2	50	115	20687	20687	20687	< 1
2	50	103	17262	17262	17262	< 1
2	50	112	19077	19077	19077	< 1
2	50	111	18739	18739	18739	< 1
3	50	103	7599	7599	7599	< 1
3	50	97	6554	6554	6554	< 1
3	50	109	8937	8937	8937	< 1
3	50	99	6621	6621	6621	< 1
3	50	118	11326	11326	11326	< 1

Fonte: Elaborado pelo autor (2022)

Quadro A.8 – Resultados do *Set Partitioning*, na terceira etapa com instâncias de tamanho 100, para zona progressiva

Tipo	<i>N</i>	<i>QP</i>	<i>RM</i>	<i>RMAP</i>	<i>RMAM</i>	<i>TMC</i>
1	100	222	20162	20133	20133	6
1	100	210	18590	18563	18563	12
1	100	204	19380	19370	19370	5
1	100	205	16513	16534	16490	11
1	100	211	19039	19032	19031	5
2	100	215	37921	37985	37905	7
2	100	203	29921	29885	29885	3
2	100	213	32420	32374	32374	3
2	100	226	40136	40086	40086	4
2	100	207	35292	35246	35246	3
3	100	199	13557	13586	13539	9
3	100	202	13415	13535	13415	8
3	100	204	12941	12923	12923	7
3	100	243	16182	16159	16159	4
3	100	223	14725	14705	14705	16

Fonte: Elaborado pelo autor (2022)

Quadro A.9 – Resultados do *Set Partitioning*, na terceira etapa com instâncias de tamanho 150, para zona progressiva

Tipo	<i>N</i>	<i>QP</i>	<i>RM</i>	<i>RMAP</i>	<i>RMAM</i>	<i>TMC</i>
1	150	336	30522	30758	30492	40
1	150	301	24536	24847	24501	70
1	150	325	26253	26217	26217	54
1	150	323	28965	29350	28924	35
1	150	311	27749	27870	27710	48
2	150	313	49737	49949	49669	13
2	150	330	53966	53891	53891	6
2	150	317	52711	52735	52655	8
2	150	320	52630	52961	52575	12
2	150	325	50596	50807	50527	17
3	150	319	20878	21109	20916	24
3	150	327	25690	25672	25672	11
3	150	317	22753	22722	22722	16
3	150	345	24578	24929	24543	18
3	150	310	23568	23537	23537	11

Fonte: Elaborado pelo autor (2022)

Quadro A.10 – Resultados do *Set Partitioning*, na terceira etapa com instâncias de tamanho 200, para zona progressiva

Tipo	<i>N</i>	<i>QP</i>	<i>RM</i>	<i>RMAP</i>	<i>RMAM</i>	<i>TMC</i>
1	200	421	38894	35132	35132	317
1	200	428	42747	36641	36507	255
1	200	416	40653	40288	40248	79
1	200	464	38548	37468	37122	282
1	200	465	48182	38972	38892	113
2	200	437	70480	70345	70305	52
2	200	430	69219	69126	69126	49
2	200	431	69379	69300	69300	46
2	200	434	73372	74908	73630	38
2	200	416	67376	67252	67252	43
3	200	445	31898	31852	31852	347
3	200	433	33499	33455	33189	113
3	200	438	31169	31128	31126	66
3	200	414	27149	28228	26950	115
3	200	421	30333	30174	30174	89

Fonte: Elaborado pelo autor (2022)

Quadro A.11 – Resultados das heurísticas de solução, na primeira etapa, para zona progressiva

Tipo	N	QP	HA	HG	HGO	HGS	HGD	RM ₁	RM ₂	HBS
1	10	21	9070	9070	8951	8951	8951	6336	6336	6336
1	10	21	13444	14777	13291	13291	13291	7754	7754	7754
1	10	21	10502	10502	10490	10490	10490	7245	7245	7245
1	10	19	10107	10107	13716	13716	13716	7990	7990	7990
1	10	18	14532	14532	12079	12079	12079	8773	8773	8773
2	10	24	13481	13481	13246	13246	13246	10110	10110	10110
2	10	26	13747	13747	14945	14945	14945	9275	9275	9275
2	10	18	10453	10453	15683	15683	15683	8223	8223	8223
2	10	26	11874	11874	11080	11080	11080	8421	8421	8421
2	10	30	17027	17027	17818	17818	17818	13120	13120	13120
3	10	25	14359	14359	14107	14107	14107	9441	9441	9441
3	10	21	15950	15950	15403	15403	15403	12926	12926	12926
3	10	22	8906	8906	11067	11067	11067	6299	6299	6299
3	10	19	12429	12442	10817	10817	10817	7186	7186	7186
3	10	20	11552	11552	11589	11589	11589	9415	9415	9415
-	-	-	6	6	9	9	9	-	-	-

Fonte: Elaborado pelo autor (2022)

Quadro A.12 – Resultados das heurísticas de solução, na segunda etapa, para zona progressiva

Tipo	N	QP	HA	HG	HGO	HGS	HGD	RM ₁	RM ₂	HBSP
1	10	20	1295	1295	1203	1203	1203	858	858	858
1	20	37	2267	2260	1846	2029	1846	1554	1554	1554
1	30	57	2523	2725	3196	3180	3118	2140	2140	2140
1	40	89	4464	4464	4100	4132	4016	3626	3626	3626
1	50	103	4588	4600	4375	4407	4375	3532	3532	3532
2	10	23	1575	1464	1580	1468	1468	1452	1452	1452
2	20	40	20496	19163	19203	18348	19203	17000	17000	17000
2	30	61	4539	4539	4820	4634	4820	4281	4281	4281
2	40	87	5171	5089	5376	5375	5376	4848	4848	4848
2	50	113	5899	5886	5575	5575	5575	5400	5400	5400
3	10	29	1069	1069	1271	1271	1271	818	818	818
3	20	43	2668	2477	2290	2290	2290	1253	1253	1253
3	30	62	2782	2464	2436	2394	2436	1699	1699	1699
3	40	86	2693	2748	3079	3079	3079	2196	2196	2196
3	50	100	3945	3817	3360	3362	3362	2983	2983	2983
-	-	-	4	4	6	5	6	-	-	-

Fonte: Elaborado pelo autor (2022)

Quadro A.13 – Resultados da primeira modelagem de cálculo de tempo, na primeira etapa, para zona sincronizada

Tipo	<i>N</i>	<i>QP</i>	<i>RM</i>	<i>TM</i>	<i>TMM</i>	<i>LI</i>	<i>GAP</i>	<i>LR</i>
1	10	21	7354	190	2	7354	0%	123
1	10	21	8473	3600	14	8449	0,28%	148
1	10	21	6061	107	< 1	6061	0%	195
1	10	19	6074	133	1	6074	0%	193
1	10	18	6302	92	1	6302	0%	123
2	10	24	7974	3600	2	6772	15,07%	184
2	10	26	5923	3600	3	5764	2,67%	287
2	10	18	5043	533	1	5043	0%	163
2	10	26	6632	191	1	6632	0%	185
2	10	30	11810	3600	4	10538	10,77%	222
3	10	25	5879	110	6	5879	0%	219
3	10	21	10426	1802	6	10426	0%	253
3	10	22	7228	3600	3	7136	1,26%	206
3	10	19	5286	336	2	5286	0%	150
3	10	20	7414	195	3	7414	0%	136

Fonte: Elaborado pelo autor (2022)

Quadro A.14 – Resultados da primeira modelagem de cálculo de tempo, na segunda etapa, para zona sincronizada

Tipo	<i>N</i>	<i>QP</i>	<i>RM</i>	<i>TM</i>	<i>TMM</i>	<i>LI</i>	<i>GAP</i>	<i>LR</i>
1	10	20	625	12	< 1	625	0,00%	59
1	20	37	1066	3600	416	606	43,15%	153
1	30	57	1589	3600	442	690	56,57%	168
1	40	89	2479	3600	3201	324	86,93%	308
1	50	103	2714	3600	412	830	69,41%	261
2	10	23	1635	3600	< 1	1380	15,59%	192
2	20	40	13400	3600	188	4515	66,30%	358
2	30	61	4720	3600	131	2278	51,73%	383
2	40	87	3698	3600	719	1278	65,44%	416
2	50	113	3645	3600	1066	1268	65,21%	540
3	10	29	943	3600	6	873	7,42%	80
3	20	43	1252	3600	465	714	42,97%	120
3	30	62	1700	3600	48	917	46,05%	199
3	40	86	2132	3600	85	1128	47,09%	310
3	50	100	2910	3600	1368	513	82,37%	383

Fonte: Elaborado pelo autor (2022)

Quadro A.15 – Resultados da segunda modelagem de cálculo de tempo, na primeira etapa, para zona sincronizada

Tipo	<i>N</i>	<i>QP</i>	<i>RM</i>	<i>TM</i>	<i>TMM</i>	<i>LI</i>	<i>GAP</i>	<i>LR</i>
1	10	21	7354	1	< 1	7354	0%	2523
1	10	21	8473	9	< 1	8473	0%	2548
1	10	21	6061	< 1	< 1	6061	0%	2328
1	10	19	6074	< 1	< 1	6074	0%	2463
1	10	18	6302	< 1	< 1	6302	0%	2523
2	10	24	7974	26	< 1	7974	0%	2561
2	10	26	5923	1	< 1	5923	0%	2687
2	10	18	5043	< 1	< 1	5043	0%	2563
2	10	26	6632	1	< 1	6632	0%	2585
2	10	30	11810	74	< 1	11810	0%	2622
3	10	25	5879	< 1	< 1	5879	0%	2352
3	10	21	10426	23	< 1	10426	0%	2652
3	10	22	7228	19	< 1	7228	0%	2339
3	10	19	5286	< 1	< 1	5286	0%	2550
3	10	20	7414	12	< 1	7414	0%	2494

Fonte: Elaborado pelo autor (2022)

Quadro A.16 – Resultados da segunda modelagem de cálculo de tempo, na segunda etapa, para zona sincronizada

Tipo	<i>N</i>	<i>QP</i>	<i>RM</i>	<i>TM</i>	<i>TMM</i>	<i>LI</i>	<i>GAP</i>	<i>LR</i>
1	10	20	625	< 1	< 1	625	0,00%	269
1	20	37	1066	3600	1	1003	5,91%	405
1	30	57	1588	3600	138	1434	9,69%	420
1	40	89	2460	3600	636	1670	32,11%	560
1	50	103	2638	3600	2259	2045	22,47%	513
2	10	23	1635	286	< 1	1635	0,00%	444
2	20	40	13400	3600	4	10018	25,23%	2491
2	30	61	4720	3600	2	4031	14,59%	635
2	40	87	3673	3600	45	2885	21,45%	668
2	50	113	3412	3600	1938	2620	23,21%	792
3	10	29	943	13	< 1	943	0,00%	323
3	20	43	1252	3600	1229	1244	0,60%	372
3	30	62	1700	3600	5	1693	0,41%	451
3	40	86	2132	3600	6	2072	2,81%	562
3	50	100	2910	3600	1689	2743	5,73%	635

Fonte: Elaborado pelo autor (2022)

Quadro A.17 – Resultados do *Set Partitioning*, na primeira etapa, para zona sincronizada

Tipo	<i>N</i>	<i>QP</i>	<i>RMAP</i>	<i>RMAM</i>	<i>RMSP</i>	<i>RMSM</i>	<i>TMC</i>
1	10	20	640	625	1360	625	< 1
1	20	37	1143	1066	2428	1066	< 1
1	30	57	1734	1623	3840	3673	< 1
1	40	89	2843	2625	5258	4810	< 1
1	50	103	3057	2868	6613	6081	1
2	10	23	1642	1635	1642	1635	< 1
2	20	40	13412	13400	22380	13400	< 1
2	30	61	4824	4755	5574	5574	< 1
2	40	87	3970	3683	5547	5297	< 1
2	50	113	4120	3668	6548	6206	1
3	10	29	948	943	1916	1916	< 1
3	20	43	1347	1252	3572	3572	< 1
3	30	62	1791	1700	5672	5672	< 1
3	40	86	2316	2156	6808	6808	< 1
3	50	100	3786	3315	8705	8705	1

Fonte: Elaborado pelo autor (2022)

Quadro A.18 – Resultados do *Set Partitioning*, na segunda etapa, para zona sincronizada

Tipo	<i>N</i>	<i>QP</i>	<i>RMAP</i>	<i>RMAM</i>	<i>RMSP</i>	<i>RMSM</i>	<i>TMC</i>
1	10	21	7354	7354	14812	7354	< 1
1	10	21	8473	8473	15074	8473	< 1
1	10	21	6061	6061	14632	6061	< 1
1	10	19	6076	6074	18123	6074	< 1
1	10	18	6302	6302	13047	6302	< 1
2	10	24	7977	7974	14011	7974	< 1
2	10	26	5923	5923	15448	5923	< 1
2	10	18	5043	5043	16488	5043	< 1
2	10	26	6632	6632	15413	6632	< 1
2	10	30	11829	11810	19547	11810	< 1
3	10	25	5898	5879	16773	5879	< 1
3	10	21	10426	10426	15384	10426	< 1
3	10	22	7228	7228	16858	7228	< 1
3	10	19	5295	5286	14916	5286	< 1
3	10	20	7414	7414	13463	7414	< 1

Fonte: Elaborado pelo autor (2022)

Quadro A.19 – Resultados do *Set Partitioning*, na terceira etapa com instâncias de tamanho 50, para zona sincronizada

Tipo	<i>N</i>	<i>QP</i>	<i>RM</i>	<i>RMAP</i>	<i>RMAM</i>	<i>TMC</i>
1	50	90	7126	7295	7195	< 1
1	50	102	6577	6883	6728	< 1
1	50	111	6375	6753	6655	< 1
1	50	117	7225	7751	7500	< 1
1	50	113	8073	8423	8242	< 1
2	50	101	11290	11551	11462	< 1
2	50	115	17457	18083	17525	< 1
2	50	103	10854	11322	11111	< 1
2	50	112	14691	14749	14682	< 1
2	50	111	13474	13631	13465	< 1
3	50	103	7718	8478	8072	< 1
3	50	97	7449	7944	7674	< 1
3	50	109	8956	11580	11098	< 1
3	50	99	6163	6457	6338	< 1
3	50	118	11363	13327	12420	< 1

Fonte: Elaborado pelo autor (2022)

Quadro A.20 – Resultados do *Set Partitioning*, na terceira etapa com instâncias de tamanho 100, para zona sincronizada

Tipo	<i>N</i>	<i>QP</i>	<i>RM</i>	<i>RMAP</i>	<i>RMAM</i>	<i>TMC</i>
1	100	222	14445	14816	14293	3600
1	100	210	13642	16338	15216	3600
1	100	204	12289	13215	12645	3600
1	100	205	12681	13044	12645	3600
1	100	211	13147	13555	12739	3600
2	100	215	28053	30127	29254	3600
2	100	203	22586	26094	24461	3600
2	100	213	23904	25705	24287	3600
2	100	226	29580	33723	31262	3600
2	100	207	25232	28004	26810	3600
3	100	199	13086	16918	14575	3600
3	100	202	14607	16996	16758	3600
3	100	204	13508	15566	14750	3600
3	100	243	16232	19122	18967	3600
3	100	223	14382	16363	15353	3600

Fonte: Elaborado pelo autor (2022)

Quadro A.21 – Resultados do *Set Partitioning*, na terceira etapa com instâncias de tamanho 150, para zona sincronizada

Tipo	<i>N</i>	<i>QP</i>	<i>RM</i>	<i>RMAP</i>	<i>RMAM</i>	<i>TMC</i>
1	150	336	24153	24671	23872	3600
1	150	301	21298	19974	18993	3600
1	150	325	21587	22028	19116	3600
1	150	323	21694	23433	21914	3600
1	150	311	21508	21789	19104	3600
2	150	313	36686	39223	37128	3600
2	150	330	37252	38327	36872	3600
2	150	317	40390	42568	40110	3600
2	150	320	39268	39686	37221	3600
2	150	325	37730	39820	38214	3600
3	150	319	21765	26741	25674	3600
3	150	327	23028	27882	26920	3600
3	150	317	22167	24913	24295	3600
3	150	345	24672	30560	28963	3600
3	150	310	23313	26811	25879	3600

Fonte: Elaborado pelo autor (2022)

Quadro A.22 – Resultados do *Set Partitioning*, na terceira etapa com instâncias de tamanho 200, para zona sincronizada

Tipo	<i>N</i>	<i>QP</i>	<i>RM</i>	<i>RMAP</i>	<i>RMAM</i>	<i>TMC</i>
1	200	421	32790	29694	27445	3600
1	200	428	33588	31231	28917	3600
1	200	416	31380	33949	29993	3600
1	200	464	33610	31426	28233	3600
1	200	465	35943	34216	31378	3600
2	200	437	63609	66303	64202	3600
2	200	430	54183	56771	53829	3600
2	200	431	61201	63402	61703	3600
2	200	434	56209	58804	55604	3600
2	200	416	52659	54327	50911	3600
3	200	445	29827	35193	32771	3600
3	200	433	31207	34684	32219	3600
3	200	438	30573	33707	32520	3600
3	200	414	28302	31196	29832	3600
3	200	421	32178	37384	35671	3600

Fonte: Elaborado pelo autor (2022)

Quadro A.23 – Resultados das heurísticas de solução, na primeira etapa, para zona sincronizada

Tipo	N	QP	HA	HG	HGO	HGS	HGD	RM ₁	RM ₂	HBSF
1	10	21	15626	8220	7354	14812	7354	7354	7354	7354
1	10	21	14020	9022	9038	12895	9038	8473	8473	8473
1	10	21	8876	6405	6061	14632	6313	6061	6061	6061
1	10	19	10669	6872	8222	18123	8222	6074	6074	6074
1	10	18	11025	6368	6368	13047	6368	6302	6302	6302
2	10	24	12363	9001	8214	12096	8214	7974	7974	7974
2	10	26	11774	9084	9153	13820	9153	5923	5923	5923
2	10	18	11315	6628	5043	16488	5043	5043	5043	5043
2	10	26	15918	7178	9048	14040	9048	6632	6632	6632
2	10	30	18393	11906	12163	14957	12163	11810	11810	11810
3	10	25	13751	6173	6442	16773	6442	5879	5879	5879
3	10	21	10675	10432	10426	15387	10426	10426	10426	10426
3	10	22	12700	8557	7769	16014	7769	7228	7228	7228
3	10	19	12660	6907	6600	14916	6600	5286	5286	5286
3	10	20	9926	7927	7927	13189	7941	7414	7414	7414
-	-	-	0	8	9	0	7	-	-	-

Fonte: Elaborado pelo autor (2022)

Quadro A.24 – Resultados das heurísticas de solução, na segunda etapa, para zona sincronizada

Tipo	N	QP	HA	HG	HGO	HGS	HGD	RM ₁	RM ₂	HBSP
1	10	20	1320	661	656	1360	848	625	625	625
1	20	37	2979	1406	1392	2194	1392	1066	1066	1066
1	30	57	3771	1984	1938	3725	1938	1589	1588	1623
1	40	89	5027	3025	2853	4377	2830	2479	2460	2625
1	50	103	7544	3405	3578	6136	3439	2714	2638	2868
2	10	23	1769	1642	1653	1653	1653	1635	1635	1635
2	20	40	23981	16611	17374	23327	17374	13400	13400	13400
2	30	61	5329	5070	4961	5574	4789	4720	4720	4755
2	40	87	6265	4138	4091	5211	4107	3698	3673	3683
2	50	113	6715	4416	4418	6473	4482	3645	3412	3668
3	10	29	1085	948	989	1912	989	943	943	943
3	20	43	2932	1558	1558	3578	1558	1252	1252	1252
3	30	62	3247	1977	1822	5672	1822	1700	1700	1700
3	40	86	5712	2486	2309	6676	2309	2132	2132	2156
3	50	100	5607	3575	3680	8259	3680	2910	2910	3315
-	-	-	0	7	7	0	7	-	-	-

Fonte: Elaborado pelo autor (2022)