

UNIVERSIDADE DE CAXIAS DO SUL
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO
MESTRADO PROFISSIONAL EM ENGENHARIA DE PRODUÇÃO

FABRÍCIO ZORRASKI

**OTIMIZAÇÃO DE ROTAS DE COLETA EM ARMAZÉNS COM COLABORAÇÃO
HUMANO-ROBÔ: UMA ABORDAGEM COMPUTACIONAL**

BENTO GONÇALVES

2024

FABRÍCIO ZORRASKI

**OTIMIZAÇÃO DE ROTAS DE COLETA EM ARMAZÉNS COM COLABORAÇÃO
HUMANO-ROBÔ: UMA ABORDAGEM COMPUTACIONAL**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Produção da Universidade de Caxias do Sul, como requisito parcial à obtenção do grau de Mestre em Engenharia de Produção.

Orientador:

Prof. Dr. Leonardo Dagnino Chiwiacowsky

BENTO GONÇALVES

2024

Dados Internacionais de Catalogação na Publicação (CIP)
Universidade de Caxias do Sul
Sistema de Bibliotecas UCS - Processamento Técnico

Z88o Zorraski, Fabrício

Otimização de rotas de coleta em armazéns com colaboração humano-robô [recurso eletrônico] : uma abordagem computacional / Fabrício Zorraski. – 2024.

Dados eletrônicos.

Dissertação (Mestrado) - Universidade de Caxias do Sul, Programa de Pós-Graduação em Engenharia de Produção, 2024.

Orientação: Dagnino Chiwiacowsky.

Modo de acesso: World Wide Web

Disponível em: <https://repositorio.ucs.br>

1. Engenharia de produção. 2. Armazéns gerais. 3. Veículos autônomos. 4. Sistemas de separação de pedidos - Automação. 5. Logística. 6. Tecnologia. I. Chiwiacowsky, Dagnino, orient. II. Título.

CDU 2. ed.: 658.75

Catalogação na fonte elaborada pela(o) bibliotecária(o)
Márcia Servi Gonçalves - CRB 10/1500

FABRÍCIO ZORRASKI

**OTIMIZAÇÃO DE ROTAS DE COLETA EM ARMAZÉNS COM COLABORAÇÃO
HUMANO-ROBÔ: UMA ABORDAGEM COMPUTACIONAL**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Produção da Universidade de Caxias do Sul, como requisito parcial à obtenção do grau de Mestre em Engenharia de Produção.

Orientador:

Prof. Dr. Leonardo Dagnino Chiwiacowsky

Aprovado em 18/12/2024

Banca Examinadora

Prof. Dr. Leonardo Dagnino Chiwiacowsky (orientador)
Universidade de Caxias do Sul

Prof. Dr. Alexandre Mesquita
Universidade de Caxias do Sul

Prof. Dr. Odacir Deonísio Graciolli
Universidade de Caxias do Sul

Prof. Dr. Nelson Hein
Universidade Regional de Blumenau

RESUMO

O processo de *picking* é uma das operações mais críticas nos Centros de Distribuição (CDs), representando uma parcela significativa dos custos operacionais, devido ao tempo despendido em deslocamentos dos operadores. A inclusão de robôs móveis, como AMRs e AGVs, vem sendo desenvolvida como uma opção para aprimorar essas operações. Esses robôs têm o potencial de reduzir deslocamentos desnecessários, se aplicados de forma eficaz, aumentando a eficiência e a produtividade dos CDs. O presente estudo concentra-se no desenvolvimento de uma ferramenta computacional que utiliza técnicas de otimização para definir rotas mais eficientes para operadores humanos em um sistema de coleta do tipo *picker-to-parts* assistido por robôs. A metodologia envolve a modelagem virtual das áreas de armazenagem dos CDs e a aplicação de uma estratégia de agrupamento para consolidar múltiplos endereços em pontos de coleta, visando a otimizar a matriz de distâncias e a movimentação dos robôs. Além disso, utiliza métodos exatos e heurísticos para solucionar problemas de roteamento, por meio da associação com os modelos do Problema do Caixeiro Viajante (PCV) e do Problema do Caixeiro Viajante Aberto (PCVA). O estudo avalia diferentes estratégias de execução das rotas de coleta, nas quais os operadores permanecem na área de *picking* enquanto os robôs retornam ao depósito central. Os resultados demonstram que, ao aplicar essas estratégias, foi possível reduzir os deslocamentos dos operadores em até 33,61%, com a estratégia de otimização alcançando 96,15% de eficácia, apresentando um gap global de 3,85% para atingir a otimização total de cada rota. A ferramenta foi validada por meio de experimentos em um cenário real, permitindo a comparação das rotas otimizadas com as rotas reais atualmente utilizadas pela empresa, mostrando um ganho na performance do processo de coleta.

Palavras-chave: armazém, otimização, veículos autônomos, coleta de pedidos, *picking*, colaboração humano-robô, AGV, AMR, GLS, *guided local search*.

ABSTRACT

The picking process is one of the most critical operations in Distribution Centers (DCs), representing a significant portion of operational costs due to the time spent on operator movement. The inclusion of mobile robots, such as AMRs and AGVs, has been developed as an option to enhance these operations. These robots have the potential to reduce unnecessary travel, when applied effectively, thereby increasing the efficiency and productivity of DCs. This study focuses on the development of a computational tool using optimization techniques to define more efficient routes for human operators in a picker-to-parts system assisted by robots. The methodology involves virtual modeling of DC storage areas and the application of a clustering strategy to consolidate multiple addresses into collection points, aiming to optimize the distance matrix and robot movements. Additionally, it employs exact and heuristic methods to solve routing problems through association with TSP and OTSP models. The study evaluates different strategies for executing collection routes, where operators remain in the picking area while robots return to the central depot. The results demonstrate that applying these strategies reduced operator movement by up to 33.61%, with the optimization strategy achieving 96.15% efficiency and a global gap of 3.85% to reach the total optimization of each route. The tool was validated through experiments in a real-world scenario, enabling a comparison between the optimized routes and the actual routes currently used by the company, showing improved performance in the collection process.

Keywords: *warehousing, optimization, order picking, routing, human-robot collaboration, AGV, AMR, GLS, guided local search.*

LISTA DE FIGURAS

Figura 1 – Fluxos e funções típicas de um Centro de Distribuição (CD)	22
Figura 2 – Exemplo de <i>layouts</i> das estruturas de armazenamentos dos produtos em armazéns	23
Figura 3 – Categorização dos sistemas de coleta de pedidos	26
Figura 4 – Exemplo de corredores transversais em armazéns com <i>layout</i> de um e dois blocos	28
Figura 5 – Exemplo de políticas de rotas <i>S-shape</i> e <i>Return</i>	29
Figura 6 – Exemplo de políticas de rotas <i>Midpoint</i> e <i>Largest Gap</i>	29
Figura 7 – Exemplo de políticas de rotas <i>Composite</i> e <i>Optimal</i>	30
Figura 8 – Exemplos de AGVs e AMRs em diferentes operações	31
Figura 9 – Contraste entre controles centralizado e descentralizado dos AGVs e AMRs .	32
Figura 10 – Coleta de pedidos realizada com auxílio de um robô	33
Figura 11 – Métodos de resolução de problemas de otimização	43
Figura 12 – Ótimos local e global no espaço de busca em um problema de minimização	47
Figura 13 – Funcionamento de uma heurística de busca local aplicada a um espaço de busca específico	48
Figura 14 – Ciclo meta-heurístico	49
Figura 15 – Deformação do espaço de soluções causada pela GLS	51
Figura 16 – Fluxogramas contendo as macro-etapas do método de trabalho	54
Figura 17 – Fluxograma do Modelo Conceitual da Ferramenta de Otimização	58
Figura 18 – Erro (%) em Função do Tempo de Execução - <i>Pr76</i>	67
Figura 19 – Erro (%) em Função do Tempo de Execução - <i>Pr136</i>	68
Figura 20 – Tempo de Execução (ms) por Número de Endereços	71
Figura 21 – Tempo de Criação da Matriz (ms) por Número de Endereços	72
Figura 22 – Estrutura dos dados utilizados para realizar as simulações	73
Figura 23 – Rota planejada pelo WMS	76
Figura 24 – Rota otimizada pela abordagem PCV	77
Figura 25 – Rotas otimizadas a partir da Estratégia 1	81
Figura 26 – Nível de otimização alcançado em cada tarefa - Estratégia 1	82
Figura 27 – Rotas otimizadas a partir da Estratégia 2	83
Figura 28 – Nível de otimização alcançado em cada tarefa - Estratégia 2	84
Figura 29 – Rota iniciando por um ponto de coleta central do armazém	85
Figura 30 – Nível de otimização alcançado em cada tarefa - Estratégia 3	86
Figura 31 – Nível de otimização alcançado em cada tarefa - Estratégia 4	87
Figura 32 – Comparação de Distâncias Totais (Operador vs. Robô)	88
Figura 33 – Proporção de deslocamentos improdutivos dos operadores humanos	88

Figura 34 – Gráfico comparativo entre as distâncias de deslocamento do operador no cenário de coletas independentes e no cenário de coletas encadeadas.	89
Figura 35 – Comparação de desempenhos relativos de cada estratégia	90
Figura 36 – Categorização de sistemas de coleta de pedidos	104
Figura 37 – Circuito de esteiras utilizado no sistema <i>pick-to-sort</i>	105
Figura 38 – Exemplo de sistema <i>A-frame</i>	106
Figura 39 – Sistemas de orientação utilizados por AGVs, vistos a partir de uma vista superior	107
Figura 40 – Sistema de orientação baseado em visão utilizado pelos AMRs	109
Figura 41 – Exemplo de árvore de recursão <i>Branch-and-Bound</i>	110
Figura 42 – Representação da execução do algoritmo de Dijkstra	118
Figura 43 – Estrutura porta <i>pallet</i> exemplificando nomenclaturas utilizados	119
Figura 44 – Interface gráfica	120
Figura 45 – Modo de construção/edição da ferramenta	121
Figura 46 – Modo de informação da ferramenta	122
Figura 47 – Análises comparativas fornecidas pela ferramenta	123
Figura 48 – Matriz de distâncias	125
Figura 49 – Estrutura de mapeamento dos endereços de armazenagem	126
Figura 50 – Estrutura de mapeamento dos endereços de armazenagem	127
Figura 51 – Pontos de coleta (em verde) posicionados no armazém	129

LISTA DE TABELAS

Tabela 1 – Resultados das Instâncias de validação executando a abordagem exata . . .	65
Tabela 2 – Resultados das Três Primeiras Execuções por Instância na Validação da Heurística	66
Tabela 3 – Resumo dos resultados obtidos das instâncias de validação executando a abordagem heurística	67
Tabela 4 – Valores Médios das Métricas de Desempenho para as Abordagens de Otimização	70
Tabela 5 – Proporção de interrupções do Método Exato de acordo com o tamanho da instância	71
Tabela 6 – Distâncias mínimas possíveis para cada tarefa de <i>picking</i>	75
Tabela 7 – Resultados consolidados do WMS para coletas independentes	76
Tabela 8 – Comparação de Distâncias (em metros) Planejadas e Otimizadas	76
Tabela 9 – Comparação das distâncias percorridas pelos operadores humanos	78
Tabela 10 – Resumo das distâncias percorridas pelo operador humano - WMS	79
Tabela 11 – Distâncias percorridas pelo robô - WMS	79
Tabela 12 – Comparação com Planejamento WMS - Estratégia 1	80
Tabela 13 – Comparação com Planejamento WMS - Estratégia 2	82
Tabela 14 – Comparação com Planejamento WMS - Estratégia 3	85
Tabela 15 – Comparação com Planejamento WMS - Estratégia 4	86
Tabela 16 – Comparação de Estratégias - Distância Durante Picking	90

LISTA DE SIGLAS

2D	Bidimensional
AGV	<i>Automated Guided Vehicle</i>
AMR	<i>Autonomous Mobile Robot</i>
AS/RS	Sistema de Recuperação e Armazenamento Automatizado
BFS	<i>Breadth First Search</i>
CD	Centro de Distribuição
DFS	<i>Depth First Search</i>
FLS	<i>Fast Local Search</i>
FO	Função Objetivo
GA	<i>Genetic Algorithm</i>
GLS	<i>Guided Local Search</i>
IA	Inteligência Artificial
JIT	<i>Just In Time</i>
LIDAR	<i>Light Detection and Ranging</i>
LIFO	<i>Last In First Out</i>
ML	Aprendizado de Máquina
PCV	Problema do Caixeiro Viajante
PCVA	Problema do Caixeiro Viajante Aberto
PCVS	Problema do Caixeiro Viajante de Steiner
PL	Programação Linear
PLIM	Programação Linear Inteira Mista
PO	Pesquisa Operacional
PRV	Problema de Roteamento de Veículos
SA	<i>Simulated Annealing</i>
SKU	<i>Stock Keeping Unit</i>
SLAM	<i>Simultaneous Localization and Mapping</i>
SPP	<i>Shortest Path Problem</i>
VLM	Módulo de Elevação Vertical
WMS	<i>Warehouse Management System</i>

SUMÁRIO

1	INTRODUÇÃO	14
1.1	OBJETIVOS DA PESQUISA	17
1.1.1	Objetivo Geral	17
1.1.2	Objetivos Específicos	17
1.2	Justificativa	17
1.2.1	Justificativa Teórica	18
1.2.2	Justificativa Prática	18
1.3	ABORDAGEM E DELIMITAÇÕES	19
1.4	Estrutura do texto	19
2	FUNDAMENTAÇÃO TEÓRICA	21
2.1	CENTROS DE DISTRIBUIÇÃO	21
2.1.1	<i>Layout</i>	23
2.1.2	Estratégias de Coleta	24
2.1.2.1	Rotas de coleta em armazéns	27
2.1.2.2	Técnicas Clássicas de Roteamento	28
2.2	Tecnologias de Movimentação Autônoma	31
2.3	Colaboração humano-robô na separação de pedidos	33
2.3.1	Problemas de decisão associados à adoção desta tecnologia	34
2.4	OTIMIZAÇÃO E MODELAGEM	35
2.4.1	Problemas de Otimização: Fundamentos e Modelagem	35
2.4.2	Modelos clássicos de otimização de rotas	37
2.4.2.1	Problema do Caixeiro Viajante (PCV)	37
2.4.2.2	Problema do Caixeiro Viajante Aberto (PCVA)	39
2.5	ESTRATÉGIAS DE SOLUÇÃO NA OTIMIZAÇÃO COMBINATÓRIA	41
2.5.1	Métodos Exatos	43
2.5.1.1	<i>Branch-and-Bound</i>	44
2.5.1.2	<i>Branch-and-Cut</i>	44
2.5.2	Métodos aproximados	45
2.5.2.1	Heurísticas	45
2.5.2.2	Processo de busca local	47
2.5.2.3	Meta-heurísticas	48
2.5.2.4	Meta-heurística <i>Guided Local Search</i> (GLS)	49
2.6	ALGORITMO DE DIJKSTRA	52
3	PROCEDIMENTOS METODOLÓGICOS	54
3.1	Macro-etapa de Planejamento	54

3.1.1	Definição do problema	54
3.1.2	Coleta de informações	56
3.1.3	Modelo Conceitual	56
3.2	Macro-etapa de Modelagem	59
3.2.1	Coleta de dados	59
3.2.1.1	<i>Layout</i> do Armazém e Estrutura de Armazenamento	59
3.2.1.2	Estratégias Atuais de Coleta	60
3.2.2	Modelagem Computacional	60
3.2.2.1	Construção da Matriz de Distâncias	60
3.2.2.2	Método de Otimização	61
3.2.2.3	Visualização dos resultados	61
3.2.2.4	Validação do Modelo	62
3.3	Macro-etapa de Experimentos de Simulação	62
3.3.1	Definição de Cenários de Avaliação	62
3.3.2	Critérios de Desempenho	62
3.3.3	Análise dos Resultados	63
3.4	Macro-etapa de Recomendações	63
4	EXPERIMENTOS E RESULTADOS	64
4.1	Experimentos de validação	65
4.1.1	Validação do Método Exato	65
4.1.2	Validação do Método Heurístico	66
4.1.3	Experimentos de Desempenho em Ambiente Real	69
4.2	Experimentos Aplicados	73
4.2.1	Cenário de Coletas Independentes: Otimização de Rotas Únicas	73
4.2.2	Cenário de Coletas Encadeadas: Otimização de Rotas Sequenciais	74
4.2.3	Margem de Otimização	75
4.2.4	Resultados	75
4.2.5	Resultados do cenário de coletas independentes (PCV)	76
4.2.6	Resultados do cenário de coletas encadeadas (PCVA)	78
4.2.6.1	Partida Ajustada: Ponto Próximo (Estratégia 1)	80
4.2.6.2	Partida Ajustada: Integra o Último Ponto à Rota Atual (Estratégia 2)	81
4.2.6.3	Adaptativa: Seleção Estratégica de Pontos Extremos (Estratégia 3)	84
4.2.6.4	Combinação Estratégica: Extremos e Ajuste da Partida (Estratégia 4)	86
4.3	Análise e Recomendações	87
5	CONCLUSÃO	92
	REFERÊNCIAS	94

APÊNDICES	101
APÊNDICE A – ESTRATÉGIAS DE ARMAZENAMENTO	102
APÊNDICE B – CATEGORIAS DE SISTEMAS DE COLETA	104
APÊNDICE C – SISTEMAS DE ORIENTAÇÃO	107
APÊNDICE D – FUNCIONAMENTO DO MÉTODO <i>BRANCH-AND-BOUND</i>	110
APÊNDICE E – FUNCIONAMENTO DOS PLANOS DE CORTE E DO MÉTODO <i>BRANCH-AND-CUT</i>	113
APÊNDICE F – PSEUDO-CÓDIGO <i>GUIDED LOCAL SEARCH</i>	115
APÊNDICE G – PSEUDO-CÓDIGO <i>DO ALGORITMO DE DIJKSTRA</i>	117
APÊNDICE H – CONCEITOS SOBRE A ESTRUTURA DE ARMAZENAMENTO	119
APÊNDICE I – INTERFACE GRÁFICA	120
APÊNDICE J – MÉTODO DE CONSTRUÇÃO E CÁLCULO DA MATRIZ DE DISTÂNCIAS	124
APÊNDICE K – MODELAGEM VIRTUAL DO ARMAZÉM	126
APÊNDICE L – CONCEPÇÃO DOS PONTOS DE COLETA	128

1 INTRODUÇÃO

Um sistema logístico tem como objetivo criar valor para o cliente por meio de atividades que garantem a entrega do produto correto, no local e momento adequado, na quantidade certa e com o menor custo possível. A ausência de armazenamento em um sistema logístico só seria viável se houvesse uma sincronização perfeita entre produção e consumo, sem variações, e se fossem utilizados meios de transporte rápidos e frequentes, para realizar a entrega de pequenas cargas diretamente ao cliente (CARVALHO, 2017).

Desta forma, os Centros de Distribuição (CDs) assumiram um papel importante dentro da cadeia de suprimentos, de modo que as organizações começaram a reconhecer que a maneira na qual os produtos (matérias-primas, produtos em processo, produtos acabados) são armazenados, interfere de forma significativa em suas operações (MANZINI, 2012). À medida que os processos de intralogística empregados na armazenagem e movimentação dos produtos são executados de forma eficiente, isso impacta na redução do *lead time* das operações, promovendo redução dos prazos de entrega e dos custos operacionais.

Conforme evidenciado por Frazelle (2016), dentre todas as operações da logística interna realizadas em armazéns tradicionais, onde os operadores coletam os itens solicitados se deslocando de prateleira em prateleira (*picker-to-parts*), o processo de separação de pedidos (*picking*) representa uma parcela significativa dos custos operacionais totais, aproximadamente 55%. Isso ocorre devido à intensa demanda por trabalho manual e mão de obra necessária para a sua execução. Além disso, como ressaltado por Fernandes (2012), o tempo gasto no processo de separação é consideravelmente afetado pelo deslocamento dentro da área de armazenagem, consumindo aproximadamente 50% do tempo total dedicado a essa operação. Isso se deve à área de armazenagem comumente ocupar uma grande extensão do CD em função da necessidade de acondicionamento dos produtos.

A eficiência logística e o processo de distribuição são vitais para a confiabilidade de uma empresa junto aos seus clientes, atuando como um diferencial competitivo, conforme destacado por Carvalho (2017). Nesse contexto, a busca por uma maior capacidade de atendimento ao cliente requer a redução de desperdícios nas operações. Portanto, na busca por eficiência e um maior controle operacional e de estoque, o uso de tecnologias como o *Warehouse Management System* (WMS) se mostram indispensáveis. Esse tipo de sistema não apenas auxilia a gestão de armazéns mas também aprimora o atendimento ao consumidor, através da gestão dos processos de entrada e saída dos produtos.

No entanto, apesar de sua importância, muitos WMS ainda não incorporam estratégias de otimização essenciais para processos críticos realizados em armazéns, como o *picking*. Tais sistemas carecem de métodos de roteamento capazes de estabelecer sequências eficazes para a coleta de produtos. Geralmente, estão limitados a estratégias baseadas em heurísticas de roteamento clássicas que, por vezes, não resultam em rotas eficientes e não se ajustam adequadamente

a todos os tipos de CDs, devido à falta de estrutura ou padronização dos *layouts* (ROODBERGEN; KOSTER, 2001b). Além disso, devido à alta rotatividade de *Stock Keeping Units* (SKUs), especialmente os SKUs relacionados a demandas sazonais, as empresas enfrentam desafios para manter a organização do armazenamento dos produtos alinhada de acordo com a curva de consumo, que está em constante mudança. Como resultado, no contexto do *picking*, os operadores podem ser direcionados a seguir sequências de coleta mais longas do que o necessário para atender às demandas dos pedidos, impactando diretamente a performance e os custos.

Nos últimos anos, o crescente volume de vendas dos *e-commerces* impulsionou o desenvolvimento de novas estratégias para otimizar o processo de separação de pedidos. Essas novas abordagens foram revisadas e discutidas de forma sistemática em artigos recentes escritos por Azadeh, Koster e Roy (2019), Boysen, de Koster e Weidinger (2019) e Gils et al. (2018), abrangendo desde adaptações no modelo tradicional de separação, como agrupamento e classificação por zona, até soluções inovadoras, como o armazenamento em prateleiras mistas e sistemas automatizados de entrega de produtos diretamente aos operadores (*parts-to-picker*).

De fato, em CDs modernos, observa-se que a automação está gradualmente substituindo as tarefas manuais envolvidas na manipulação de materiais, com o propósito de reduzir o tempo gasto pelos operadores humanos em atividades manuais e de baixo valor agregado. Diante deste cenário, surgiu recentemente um novo formato das operações de *picking*, onde o separador de pedidos é acompanhado por um veículo automatizado, geralmente um *Autonomous Mobile Robot* (AMR) ou *Automated Guided Vehicle* (AGV), ao longo do processo de coleta de pedidos (FRAGAPANE et al., 2021).

Este sistema, denominado “sistema de separação de pedidos assistido/auxiliado por AMRs ou AGVs”, conforme descrito por Fragapane et al. (2021), é o foco principal do presente estudo. Pretende-se, portanto, a partir da análise do contexto operacional observado em CDs, desenvolver uma ferramenta que auxilie na construção de sequências de coleta mais eficientes dentro de um ambiente de colaboração entre humanos e robôs. Especificamente, aborda-se o cenário em que operadores humanos, seguindo o método tradicional de coleta *picker-to-parts*, são apoiados por AMRs ou AGVs. Estes robôs não apenas acompanham os operadores humanos durante a coleta de todos os itens, mas também assumem a responsabilidade de transportar os itens coletados até o destino final.

O desafio de determinar uma sequência eficiente para o processo de coleta de itens é identificado como um problema de otimização e é reconhecido pela sua alta complexidade devido à grande quantidade de rotas possíveis. Esse grande número de alternativas é resultado do fenômeno conhecido como explosão combinatória, que é caracterizado pelo aumento exponencial do número de soluções possíveis em situações onde as variáveis podem ser combinadas de diversas formas. Esse crescimento exponencial torna oneroso o processo de identificação da solução ótima, exigindo estratégias de resolução variadas, que podem incluir métodos exatos, heurísticos ou uma combinação dos dois (PAPADIMITRIOU; STEIGLITZ, 1998).

Diante da complexidade em lidar com problemas que envolvem uma grande quantidade de possíveis soluções, surge a importância da Pesquisa Operacional (PO) como campo de conhecimento fundamental na resolução desses desafios complexos encontrados em diversas áreas, incluindo os enfrentados na separação de pedidos em centros de distribuição. A área de PO é constituída por um conjunto de abordagens científicas com a finalidade de otimizar a eficiência das decisões em organizações e processos, levando em consideração a escassez de recursos disponíveis (WINSTON; GOLDBERG, 2004). A metodologia empregada por essa disciplina tem como objetivo a alocação ótima dos recursos disponíveis. Portanto, a problemática de identificação da sequência ideal da coleta de itens de pedidos pode ser associada ao problema de formação de rotas, podendo ser explorada por meio da otimização combinatória, uma subárea específica da PO.

Em geral, problemas de otimização envolvem maximizar ou minimizar uma função sobre um determinado domínio. Na otimização combinatória, esse domínio é discreto e finito, o que, teoricamente, possibilita testar todas as possibilidades em busca da melhor solução. No entanto, aplicar essa abordagem mais simples geralmente é impraticável, pois o tamanho do domínio pode tornar esta estratégia proibitiva. Portanto, torna-se necessário empregar técnicas mais sofisticadas para encontrar soluções de valor ótimo (COOK et al., 1998).

Em cenários que envolvem um grande número de possibilidades, as heurísticas se destacam por sua agilidade e flexibilidade. De modo geral, elas permitem a obtenção de soluções em tempos mais viáveis quando comparadas aos métodos exatos, que geralmente são mais demorados (REEVES, 1995). Além disso, Reeves (1995) ressalta que a flexibilidade das heurísticas possibilita lidar com situações de maior complexidade, comumente observadas em problemas reais, permitindo sua aplicação prática em contextos onde soluções subótimas são aceitáveis em troca de maior rapidez, diferentemente das soluções exatas, cuja obtenção tende a ser mais demorada.

Desta forma, em busca de aumentar a produtividade operacional de centros de distribuição, considerando a importância e o elevado custo associados ao processo de coleta de produtos em armazéns tradicionais, foi desenvolvida uma ferramenta computacional para aprimorar a operação de separação de pedidos (*picking*). A ferramenta foi projetada para uso em cenários colaborativos, onde humanos e robôs trabalham de forma integrada, sendo desenvolvida como um recurso externo a ser empregado de forma associada aos sistemas de gestão do armazém. Ela permite a otimização de roteiros de separação com base em diferentes estratégias operacionais, fornecendo resultados que podem ser utilizados por sistemas de gestão de armazéns. A eficácia e aplicabilidade da ferramenta foram testadas em um cenário real, validando seu desempenho.

1.1 OBJETIVOS DA PESQUISA

A presente pesquisa possui os objetivos a seguir definidos.

1.1.1 Objetivo Geral

Esta pesquisa tem como objetivo geral desenvolver uma ferramenta computacional de otimização de *picking* colaborativo humano-robô.

1.1.2 Objetivos Específicos

Para atender ao objetivo geral, foram formulados os seguintes objetivos específicos:

- a) Possibilitar a representação das características físicas e operacionais do ambiente de armazenamento a ser avaliado;
- b) Desenvolver métodos computacionais capazes de otimizar rotas para a separação colaborativa de pedidos entre operadores humanos e robôs;
- c) Validar a ferramenta desenvolvida por meio de aplicação prática em um ambiente real de *picking*;
- d) Analisar diferentes estratégias de operação com base na avaliação dos resultados obtidos em cenários reais de operação de *picking*.

1.2 JUSTIFICATIVA

Com o avanço do comércio eletrônico, as empresas enfrentam o desafio de armazenar e gerenciar uma grande variedade de produtos, lidando diariamente com volumes consideráveis e variáveis de pedidos. Nesse contexto, o processo de coleta de pedidos, essencial para atender a essa demanda crescente, torna-se cada vez mais trabalhoso e complexo para os armazéns. Frequentemente, esse processo se caracteriza por tarefas repetitivas, longos deslocamentos, exposição a condições ergonômicas adversas e dependente de mão de obra qualificada e disposta para trabalhar em diferentes turnos, um requisito que muitas vezes é difícil de ser atendido. Além disso, encontrar terrenos disponíveis para a construção de armazéns, idealmente localizados próximos aos centros de demanda, tornou-se um desafio, seja pela escassez de opções ou pelo custo, o que tem levado muitos armazéns a operarem de forma ininterrupta, 24 horas por dia, 7 dias por semana, para garantir um nível de serviço logístico ¹ adequado aos clientes (AZADEH; KOSTER; ROY, 2019).

Diante deste cenário, os CDs, e em especial o processo de coleta de pedidos, se destacam como alvos prioritários para implementar soluções de automação (AZADEH; KOSTER; ROY,

¹ O nível de serviço logístico mede a eficiência da empresa em gerenciar transporte, armazenamento, distribuição e entrega. Um alto nível indica eficiência e competitividade no atendimento ao mercado, enquanto um baixo nível leva à perda de oportunidades, desperdício de recursos e insatisfação dos clientes (TOTVS, 2024).

2019). Contudo, apenas adotar recursos de automação não garante aumento de produtividade ou eficiência se as atividades não forem adequadamente analisadas e projetadas. Assim, a otimização das estratégias de automação adotadas se torna essencial para garantir resultados positivos. Portanto, o tema deste trabalho é de particular relevância no contexto atual da logística e cadeia de suprimentos, onde a eficiência operacional e a agilidade logística são cruciais para atender às demandas de um mercado cada vez mais dinâmico e orientado pela rapidez.

1.2.1 Justificativa Teórica

A integração de AGVs ou AMRs no processo colaborativo de coleta de pedidos em armazéns representa um avanço na logística moderna e no gerenciamento da cadeia de suprimentos, visando a aprimorar a eficiência operacional e a ergonomia do trabalho. Essa abordagem de automação é uma tecnologia recente que está se tornando popular na prática, muito devido à sua simplicidade e flexibilidade. Ela oferece um potencial de aprimoramento ao processo de coleta de pedidos, ao tirar vantagem da flexibilidade, autonomia e consistência dos robôs, juntamente com a capacidade de discernimento humano. Embora já existam estudos abordando a colaboração humano-robô em armazéns, trabalhos específicos sobre técnicas de otimização nesse contexto ainda são relativamente limitados, conforme indicado por (AZADEH; KOSTER; ROY, 2019). Diante desse cenário, este estudo busca contribuir para a ampliação desse conhecimento, explorando dinâmicas operacionais e desenvolvendo estratégias computacionais que melhorem a eficiência desse processo colaborativo.

1.2.2 Justificativa Prática

A implementação e otimização do recurso de automação nos processos colaborativos de coleta de pedidos em armazéns oferecem diferentes benefícios práticos. Melhoram a eficiência operacional e reduzem as exigências físicas e os riscos ergonômicos aos quais estão expostos os encarregados pelo processo. A integração de AGVs ou AMRs à tarefa de *picking* fornece suporte aos operadores humanos pelo acompanhamento e transporte, de forma autônoma, dos itens coletados ao longo do trajeto de coleta e, posteriormente, para as áreas de embalagem ou fracionamento. Estes recursos de automação eliminam a necessidade de múltiplas viagens improdutivas dos operadores ao depósito central, permitindo a eles focar na coleta contínua de pedidos, o que possibilita finalizar mais pedidos em menos tempo. Além disso, a integração entre humanos e robôs otimiza os fluxos de trabalho, reduz os tempos do ciclo de coleta e minimiza tanto o desgaste físico dos operadores quanto seus períodos de improdutividade, causados por conta dos deslocamentos desnecessários. Nesse contexto, investigar estratégias para otimizar esse processo tem relevância prática, pois direciona a aplicação dessas tecnologias em um setor crítico da economia, promovendo cadeias de suprimentos mais ágeis e eficientes por meio da integração tecnológica.

1.3 ABORDAGEM E DELIMITAÇÕES

Esta pesquisa foi conduzida utilizando a metodologia de modelagem e simulação, estruturada em três fases principais: a concepção ou formulação do modelo, a implementação do modelo e a experimentação e análise dos resultados. Ao longo do processo, essas fases principais foram subdivididas em etapas mais detalhadas. De acordo com Hollocks (2003), a simulação computacional é definida como uma abordagem da pesquisa operacional que permite a criação de um modelo computacional de um segmento específico do mundo real. Esta técnica possibilita a realização de experimentos com o modelo para prever os resultados na realidade. Adicionalmente, foi realizada uma investigação na área de otimização combinatória, que é um ramo da PO, com o objetivo de identificar modelos e técnicas que direcionem o processo de interesse à minimização dos custos associados aos deslocamentos de *picking*, considerando o processo colaborativo entre humanos e robôs.

Em relação às delimitações, este estudo foca exclusivamente na otimização de processos de coleta de pedidos assistidos por AGVs ou AMRs, resolvendo o problema do roteamento para um único operador em um contexto de colaboração humano-robô. A otimização concentra-se estritamente na definição da rota, sem levar em conta outras possíveis estratégias de melhoria, tais como a reorganização do *layout*, a implementação de técnicas de agrupamento de pedidos ou a utilização de estratégias de armazenamento específicas. Além disso, o estudo não contempla o dimensionamento das frotas de AGVs ou AMRs, partindo do pressuposto de que, sempre que necessário, ao menos um robô estará à disposição do operador humano, eliminando a necessidade de considerar este atraso.

Como a ferramenta foi desenvolvida para ser um recurso externo associado aos sistemas de gestão do armazém, ela não irá exigir conhecimento ou acesso direto aos dados dos produtos a serem coletados ou dos estoques disponíveis. Portanto, outra limitação é que a ferramenta depende do recebimento prévio de uma lista de endereços contendo os locais dos produtos a serem coletados na tarefa de separação em avaliação. Dessa forma, a limitação de capacidade de carga dos robôs não é um fator considerado no presente estudo, ficando sob a responsabilidade do sistema de gestão que fornece a tarefa de separação monitorar e gerenciar esse aspecto.

1.4 ESTRUTURA DO TEXTO

Este trabalho está organizado em cinco capítulos, seguidos pelas referências e apêndices. A seguir, detalha-se o conteúdo e os objetivos de cada capítulo, proporcionando uma visão geral da estrutura do trabalho.

O Capítulo 1 de introdução, consiste no contexto geral da pesquisa, focando nos desafios logísticos enfrentados pelos centros de distribuição, especialmente relacionados à operação de coleta de pedidos. O problema do roteamento de separação de pedidos em ambientes colaborativos humano-robô é introduzido, destacando a relevância da otimização deste processo

para reduzir custos e melhorar a eficiência operacional. O capítulo também define os objetivos da pesquisa, justifica o trabalho sob aspectos teóricos e práticos, e descreve a abordagem e as delimitações do estudo.

O Capítulo 2 oferece uma revisão dos conceitos fundamentais necessários para a compreensão do estudo. Inclui uma discussão sobre os centros de distribuição, *layout* de seus armazéns, e as diferentes estratégias clássicas utilizadas na coleta de pedidos. Além disso, são discutidas as tecnologias de movimentação autônoma e a colaboração humano-robô na separação de pedidos, com ênfase nas abordagens de otimização aplicáveis, como o Problema do Caixeiro Viajante (PCV) e o Problema do Caixeiro Viajante Aberto (PCVA). O capítulo conclui com a apresentação das estratégias de solução para problemas de otimização combinatória, como a utilização de métodos exatos, heurísticas e meta-heurísticas.

O Capítulo 3 apresenta os procedimentos metodológicos empregados para abordar o problema estudado. A pesquisa é dividida em quatro macro-etapas: (i) Planejamento, onde se define o problema, coleta de informações macro e o modelo conceitual; (ii) Modelagem, que abrange a coleta de dados e a modelagem computacional; (iii) Experimentos de Simulação, onde são definidos os cenários de avaliação, critérios de desempenho e a análise dos resultados; e (iv) Recomendações, onde se define recomendações baseados nas experimentações.

O Capítulo 4 apresenta os experimentos realizados para validar e testar a eficácia da ferramenta desenvolvida. São analisados os resultados da validação dos métodos exato e heurístico, além dos experimentos de desempenho em ambiente real. As simulações para diferentes estratégias de otimização de rotas em um cenário real de operação são detalhadas, incluindo comparações com as rotas atualmente utilizadas pela empresa. Por fim, discute-se as implicações das estratégias de otimização implementadas.

O Capítulo 5 retoma o objetivo da pesquisa e destaca as principais conclusões e contribuições do trabalho. São apresentadas as implicações dos resultados, as limitações do estudo e sugestões para futuras pesquisas no campo da otimização em ambientes de colaboração humano-robô.

Por fim, são apresentadas as referências bibliográficas utilizadas na pesquisa, seguidas pelos apêndices, que fornecem informações adicionais sobre a fundamentação teórica, detalhes técnicos dos algoritmos utilizados e a execução de procedimentos complementares para o desenvolvimento da ferramenta de otimização.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo é dedicado à apresentação dos conceitos teóricos fundamentais relacionados aos tópicos relevantes para a pesquisa. São apresentados conceitos relacionados às características físicas e operacionais de armazéns, aos recursos de movimentação autônoma e às abordagens de otimização que surgem como alternativas de uso nesse contexto.

2.1 CENTROS DE DISTRIBUIÇÃO

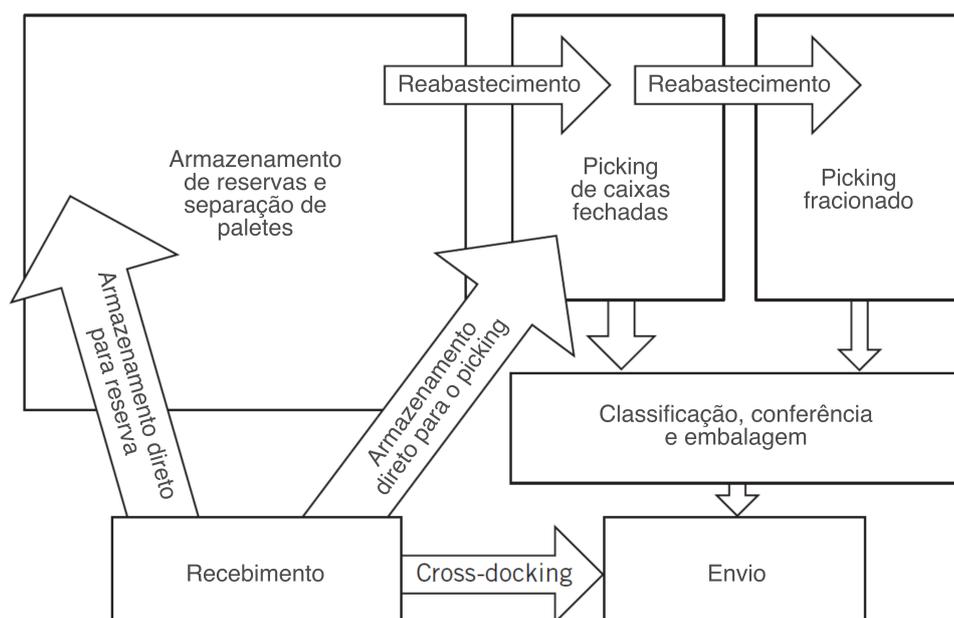
Os centros de distribuição ou armazéns atuam como a ligação física entre fornecedores e consumidores, sendo originados da necessidade de manter estoques devido aos padrões distintos entre produção e consumo dos produtos ao longo do tempo. Enquanto a demanda pelo produto é contínua, a produção frequentemente acontece em lotes, seja de encomenda ou de produto (CARVALHO, 2017). Além disso, há diversas razões que levam as empresas a construírem um armazém, conforme apontado por Hompel e Schmidt (2007):

- a) Diminuir custos de transporte: É uma das vantagens do controle e da manutenção de estoques, pois permite um uso eficaz da capacidade de carga dos veículos destinados ao transporte de produtos. Lidar com um menor número de cargas maiores geralmente é mais conveniente do que lidar com um número maior de cargas pequenas;
- b) Aprimorar a eficiência logística: A entrega de produtos corretos, no local correto e no prazo adequado é a prioridade do cliente. Estimar com precisão a demanda futura de produtos é uma tarefa desafiadora, de modo que a estratégia ideal passa a ser manter um estoque mínimo, assegurando disponibilidade de produtos e prontidão para entrega. Isso se torna ainda mais crucial em cenários com longas distâncias entre a origem e o destino dos produtos;
- c) Assegurar a eficiência operacional: Cadeias de produção desenvolvidas para implementar o sistema *Just In Time* (JIT), com a gestão de estoques mínimos, podem ser impactadas por perturbações. O controle e a manutenção de estoques desempenham um papel crucial ao garantir um fluxo contínuo ao longo de toda a cadeia de suprimentos;
- d) Agregar valor às atividades: Operações realizadas no armazém, como personalização de pedidos, manipulação, sequenciamento, preparação, pequenas montagens e desmontagens, gestão de retornos e devoluções, embalagem, etiquetagem, *cross-docking*, entre outras, incorporam valor aos produtos armazenados e manipulados;
- e) Equilibrar quantidades necessárias e entregues: Embora o mercado global tenha adotado uma abordagem de produção impulsionada pela demanda (sistema puxado), diversas indústrias ainda necessitam produzir lotes prévios para gerenciar variações sazonais ou evitar períodos de inatividade entre temporadas de trabalho.

Os CDs são o último ponto na cadeia de suprimentos antes de chegar ao cliente final. De acordo com uma perspectiva mais contemporânea, os armazéns não são mais apenas um ponto irrelevante do processo, mas sim uma peça fundamental em toda a cadeia de suprimentos.

De modo geral, para gerenciar a entrada e saída de produtos em um CD, são necessárias pelo menos cinco atividades principais. A Figura 1 ilustra esquematicamente a ordem em que normalmente os processos são realizados: recebimento, armazenamento, *picking*, consolidação ou embalagem e, por fim, expedição.

Figura 1 – Fluxos e funções típicas de um CD



Fonte: Adaptado de Tompkins et al. (2010).

Recebimento é o primeiro processo realizado, iniciando-se com a chegada e descarregamento das mercadorias enviadas pelo fornecedor. Geralmente, é realizada uma inspeção e feito um controle dos itens entregues para garantir que a quantidade e a qualidade dos produtos entregues estejam conforme solicitado. Além disso, em certos casos, há atividades de reacondicionamento para novas embalagens daqueles itens que chegam em embalagens de terceiros ou a granel, seguida da respectiva identificação por meio do uso de etiquetas.

Armazenamento é o processo realizado após o recebimento, envolve a movimentação e disposição precisa dos produtos nos locais apropriados dentro da área de armazenagem do armazém. Durante esse processo, a decisão mais importante é determinar onde e como os materiais serão armazenados, já que isso afeta diretamente as etapas seguintes, como a separação de pedidos.

A separação de pedidos é considerada a operação que apresenta maior impacto operacional e será discutida com maiores detalhes na Seção 2.1.2. Por ora, a separação de pedidos pode ser definida como o processo de coletar produtos do estoque para atender a pedidos específicos de clientes (MANZINI, 2012).

As atividades de consolidação e embalagem envolvem o processo de agrupamento e a adequada preparação dos diferentes produtos coletados para envio, frequentemente reunidos em momentos distintos de acordo com a estratégia de separação adotada, mas que constituem parte do pedido de um mesmo cliente.

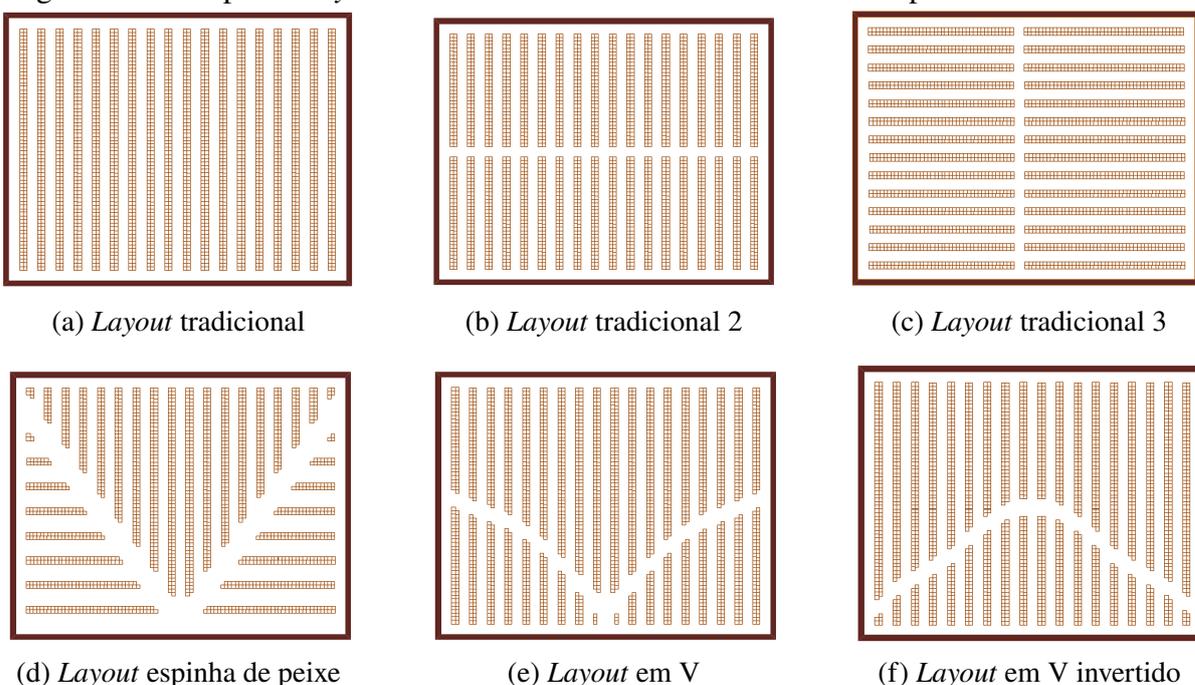
O último estágio é a expedição, na qual, além da saída física dos produtos do armazém, podem ocorrer também atividades como validação dos volumes a serem expedidos, preparação de documentos fiscais e carregamento dos itens nos veículos de transporte.

Em alguns armazéns, ainda ocorre o processo de *cross-docking*. Neste caso, o armazém opera como uma plataforma de transição para mercadorias já prontas para seu destino final, ou seja, os produtos recebidos não são armazenados, mas sim direcionados imediatamente para o processo de expedição.

2.1.1 Layout

A eficiência das operações nos armazéns é fortemente impactada pelo *layout* das estruturas de armazenamento (Figura 2), que também é crucial para o aproveitamento do espaço físico do CD de forma eficaz. Em operações de coleta, é preferível incluir corredores transversais (Figuras 2b e 2c), oferecendo maior flexibilidade para as rotas de separação, pois possibilitam caminhos mais curtos, reduzindo assim o tempo gasto no processo. No entanto, é importante observar que isso aumentará o espaço ocupado pelo armazém à medida que mais corredores transversais forem adicionados (ROODBERGEN; KOSTER, 2001a).

Figura 2 – Exemplo de *layouts* das estruturas de armazenamentos dos produtos em armazéns



Fonte: Adaptado de Karasek (2013)

Os *layouts* em formato de V e Espinha de Peixe apontam uma redução de 10% a 20% na

distância percorrida durante a coleta de pedidos, em comparação com os *layouts* que possuem corredor central (GUE; MELLER, 2009). Adicionalmente, o *layout* em V Invertido (Figura 2f) proporciona uma redução adicional de 3% na distância percorrida (GUE; IVANOVIC; MELLER, 2012).

No entanto, conforme Karasek (2013) observou, os benefícios de eficiência na coleta de produtos, resultantes da otimização do *layout* das prateleiras, dependem da interação entre o design do armazém e diversos fatores, incluindo: i) estratégias de roteamento para coletas, ii) disponibilidade de espaço nos corredores e cruzamentos, iii) equipamentos utilizados pelos operadores, como por exemplo, empilhadeiras, paleteiras ou carrinhos, iv) tamanho dos itens coletados e sua frequência na composição de pedidos, v) políticas de distribuição de pedidos e vi) estratégias de armazenamento dos produtos. Dentre os fatores destacado por Karasek (2013), as estratégias empregadas na definição das rotas de coleta são o foco do presente estudo.

2.1.2 Estratégias de Coleta

Dentro das diversas operações executadas em um CD, o processo de coleta de pedidos, também conhecido como *picking*, destaca-se como a principal atividade. Conforme já destacado anteriormente (FRAZELLE, 2016), o processo de coleta representa 55% dos custos operacionais, sendo que 50% do tempo gasto neste processo está relacionado aos deslocamentos realizados dentro da área de armazenamento/coleta (FERNANDES, 2012). Portanto, é crucial buscar uma alta eficiência operacional nesse processo. A coleta de produtos envolve a identificação, seleção e retirada de um item do seu endereço de armazenamento.

Uma ordem de coleta, também conhecida como *Order Picking*, consiste em uma lista de produtos solicitados por um ou mais clientes, na qual cada registro representa um SKU único, juntamente com a quantidade desejada (KOSTER; DUC; ROODBERGEN, 2007). Normalmente, cada ordem requer apenas uma pequena fração do total de produtos disponíveis no armazém. A eficiência do processo de coleta está diretamente ligada à política ou estratégia adotada para a sua realização, que deve levar em conta o perfil, as características dos pedidos feitos pelos clientes e os tipos de produtos envolvidos. As políticas ou estratégias de coleta de pedidos adotadas são fundamentais para direcionar como as ordens de coleta devem ser organizadas e estruturadas, definindo assim a forma como serão executadas pelos operadores em etapas subsequentes. Conforme Carvalho (2017), as estratégias mais comuns para executar a coleta dos produtos são assim identificadas:

- a) *Basic Order Picking* (*picking* discreto);
- b) *Batch Picking* (*picking* por lotes);
- c) *Zone Picking* (*picking* por zona);
- d) *Wave Picking* (*picking* por ondas).

No método de *picking* discreto, o operador coleta um produto por vez, na quantidade especificada em cada um dos pedidos individuais realizados pelos clientes. Isso implica em deslocar-se por todas as localizações dos produtos indicados no pedido, até sua conclusão. Dessa forma, ao organizar o *layout* do armazém e definir as estratégias de armazenagem (Apêndice A), é fundamental considerar a proximidade entre os produtos de alta demanda, de preferência armazenando-os próximos à área de expedição ou mesas de embalagem, visto que não é necessário realizar consolidações de diferentes ordens de coleta. Embora esse método seja simples e minimize erros, é menos produtivo devido ao tempo adicional necessário para completar cada pedido, resultante dos extensos deslocamentos. Geralmente ele é indicado para ser usado em operações com pedidos que possuem muitos itens, uma vez que reduz a probabilidade de erros ao se coletar um pedido de cada vez e também não sobrecarrega a área de consolidação (KOSTER; DUC; ROODBERGEN, 2007; RICHARDS, 2018).

Já no *picking* por lotes, várias ordens de separação são agrupadas em conjuntos (lotes) para serem realizadas simultaneamente, ou seja, o operador irá coletar produtos de diferentes pedidos ao mesmo tempo. Após a coleta dos itens do lote, os produtos são separados e organizados de acordo com cada pedido individualmente. Usualmente, as ordens de coleta são combinadas em lotes com base em critérios específicos, como localização dos itens no armazém, similaridade dos produtos ou horários de entrega. Normalmente, esse método confere uma boa eficiência ao processo de separação, dado que o deslocamento do operador pelo armazém é reduzido, já que ele consegue coletar produtos de diferentes pedidos enquanto passa pelas diferentes partes do armazém, eliminando a necessidade de retornar a essas regiões separadamente para cada pedido. Geralmente, é utilizado em operações com pedidos que possuem poucos itens (KOSTER; DUC; ROODBERGEN, 2007).

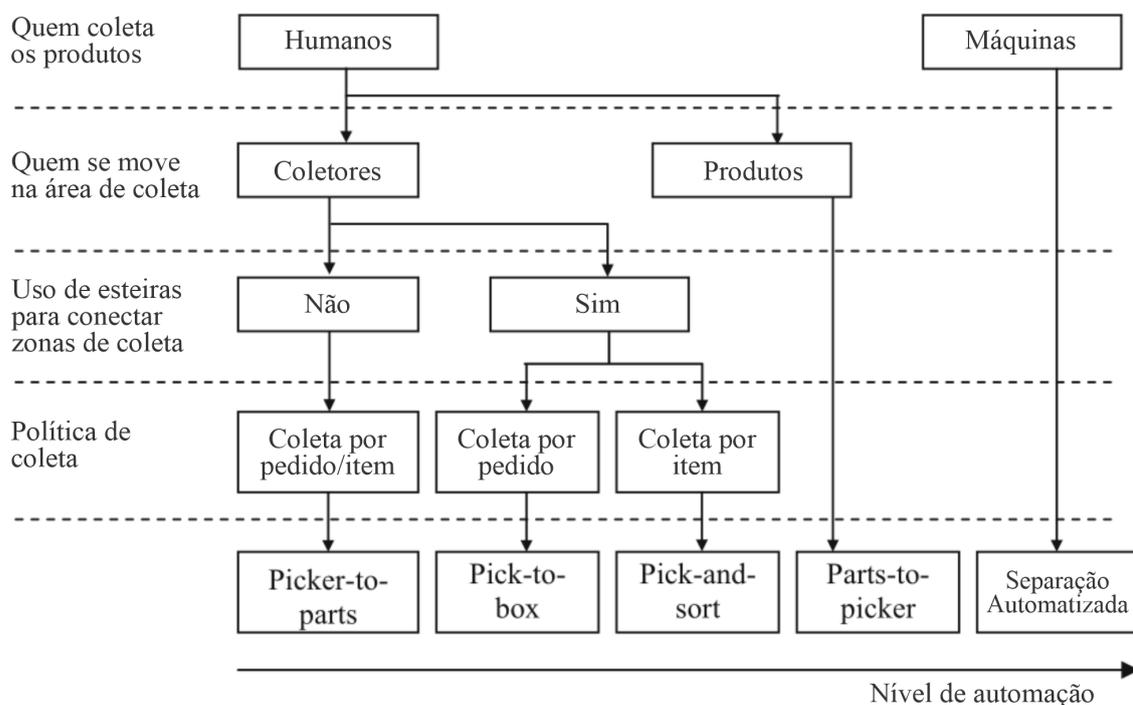
No *picking* por zona, os endereços de armazenamento do CD são divididos em diferentes zonas ou áreas, levando em consideração as características dos produtos. Cada operador é designado para trabalhar em uma ou mais zonas específicas, a fim de coletar os produtos associados a essas áreas. Em seguida, os pedidos são divididos em diferentes ordens de *picking*, de modo que cada uma corresponde às diferentes zonas em que se encontram os produtos pertencentes a esses pedidos. As vantagens do zoneamento incluem uma redução na área de deslocamento para cada operador, redução de congestionamento nos corredores, e familiarização dos endereços e produtos pertencentes à zona. Uma desvantagem é a necessidade de consolidar novamente os pedidos, uma vez que eles foram divididos para a realização do processo de separação. Uma alternativa para lidar com isso é adotar uma abordagem de coleta progressiva, na qual um operador inicia o pedido. Ao completar sua parte, o recipiente com os produtos coletados é passado para o próximo operador, que continua a montagem do pedido. Assim, um pedido só é finalizado após ter passado por todas as zonas necessárias, eliminando a necessidade de realizar sua consolidação (KOSTER; DUC; ROODBERGEN, 2007; RICHARDS, 2018).

No *picking* por ondas, os pedidos são acumulados ao longo de um período e liberados

simultaneamente, geralmente com base em critérios de similaridade na entrega, tais como destino, consolidação de carga, horário de saída ou empresa responsável pelo transporte dos volumes. Durante essa onda, os operadores coletam itens de vários pedidos em diferentes áreas do armazém, realizando uma pré-separação dos produtos de cada pedido. Após essa etapa, os produtos passam por um processo de consolidação, onde ocorre a verificação das quantidades e a alocação dos produtos em volumes separados e individualizados para cada um dos pedidos. Um dos benefícios está no aumento da eficiência ao coletar vários itens de múltiplos pedidos, simultaneamente. Adicionalmente, se a liberação das ondas for realizada de maneira alinhada com os embarques comuns, isso pode garantir uma melhor organização no setor de expedição, evitando a coleta fragmentada de pedidos, o que ocuparia o espaço físico até que esses pedidos fossem despachados (FRAZELLE, 2016).

Existem diversas categorias de sistemas de coleta. Para categorizá-los, foi adotada a classificação proposta por Dallari, Marchet e Melacini (2009), a qual se baseia em quatro critérios distintos, permitindo a diferenciação entre cinco categorias de sistemas, conforme mostrado na Figura 3. Os critérios são: i) quem é responsável pela coleta de produtos (humanos ou máquinas), ii) quem se desloca na área de coleta (coletores ou produtos), iii) é feito o uso, ou não, de esteiras para movimentação dos materiais entre zonas de coleta e iv) política/estratégia de coleta. Com base nesses critérios, as categorias resultantes são: *picker-to-parts*, *pick-to-box*, *pick-and-sort*, *parts-to-picker* e separação totalmente automatizada.

Figura 3 – Categorização dos sistemas de coleta de pedidos



Fonte: Adaptado de Dallari, Marchet e Melacini (2009).

No sistema *picker-to-parts*, os operadores se deslocam até os endereços específicos dos produtos a serem coletados para atender os pedidos. Este é o método mais comum, geralmente

são manuais e sem automação (KOSTER; DUC; ROODBERGEN, 2007). Considerando o foco deste trabalho no sistema *picker-to-parts*, apenas ele será explicado em detalhes. Informações sobre os demais sistemas podem ser encontradas no Apêndice B.

Além das categorias mencionadas, o aumento das vendas nos *e-commerces* impulsionou o estudo e desenvolvimento de novas estratégias. Isso é evidente nas revisões sistemática realizadas por Azadeh, Koster e Roy (2019), Boysen, de Koster e Weidinger (2019) e Gils et al. (2018). Nos últimos anos, uma categoria emergente começou a ser incorporada, na qual humanos e robôs colaboram no processo de *picking*, estabelecendo uma parceria humano-robô. Nesse contexto, mantendo-se dentro da classificação *pick-to-parts*, os operadores humanos recebem suporte de AGVs ou AMRs durante a coleta de pedidos. Esse sistema será detalhado na Seção 2.3.

2.1.2.1 Rotas de coleta em armazéns

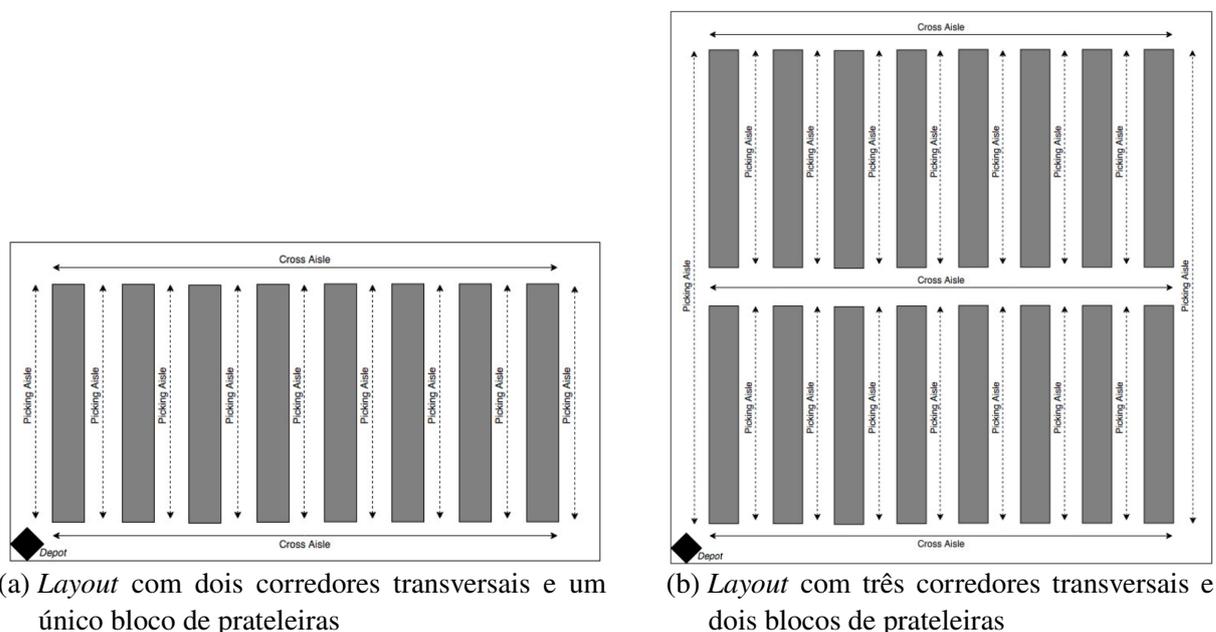
Dentre as atividades que podem aumentar o tempo de realização do processo de coleta dos produtos, o deslocamento realizado pelos operadores é a principal, podendo ser responsável por mais de 50% do tempo total de coleta (FERNANDES, 2012). Por conta desta relevância, é fundamental o uso de técnicas que auxiliem na definição de estratégias de roteamento eficientes.

O objetivo do estudo de estratégias ou métodos de roteamento é determinar o percurso do operador ao longo das posições de armazenamento a serem visitadas em cada coleta de pedido, visando a minimizar a distância percorrida ou o tempo total do percurso. Esse objetivo busca reduzir a fadiga dos operadores, diminuir o tempo necessário para a execução das atividades e, conseqüentemente, aumentar a eficiência da operação. Este estudo é conceitualmente definido como o problema de roteamento dos operadores em um armazém, podendo ser associado com o Problema do Caixeiro Viajante (PCV), detalhado na Seção 2.4.2.1. Alguns autores também associam este estudo como o Problema do Caixeiro Viajante de Steiner (PCVS), devido a duas condições: alguns endereços não precisam ser visitados, enquanto outros endereços podem ser visitados mais de uma vez (KOSTER; DUC; ROODBERGEN, 2007; KULAK; ŞAHIN; TANNER, 2012). Além disso, o problema de roteamento em armazéns pode ser relacionado com o Problema de Roteamento de Veículos (PRV) (GÓMEZ; GALAFASSI; CHIWIACOWSKY, 2012), quando para a definição das rotas de coleta existe uma restrição de capacidade máxima de peso a ser respeitada em cada rota de separação.

Os estudos de roteirização em armazéns geralmente se dividem em dois grupos principais: os referentes a armazéns com *layout* de bloco único (Figura 4a), caracterizados por possuírem um único bloco de prateleiras paralelas de comprimentos iguais, sendo os mais estudados pelos pesquisadores; e os relacionados a armazéns com dois ou mais blocos de prateleiras (*layout* de múltiplos blocos) (Figura 4b).

O problema de determinar o melhor roteiro de coleta em armazéns não é uma tarefa trivial, principalmente quando o armazém apresenta um *layout* com múltiplos blocos. Esse tipo de problema, é classificado como NP-difícil, o que significa que possui ordem de complexidade

Figura 4 – Exemplo de corredores transversais em armazéns com *layout* de um e dois blocos



Fonte: Adaptado de Rensburg (2019)

exponencial (PAPADIMITRIOU; STEIGLITZ, 1998).

Quanto aos métodos de roteirização em armazéns, eles se dividem em duas categorias distintas. A primeira categoria inclui os métodos focados em identificar as rotas mais eficientes, utilizando abordagens ótimas ou quase ótimas, enquanto a segunda categoria compreende as técnicas clássicas de roteamento. Esta segunda categoria aplica princípios fixos mesmo em diferentes configurações e combinações de produtos a serem coletados, visando à criação de rotas viáveis, embora não necessariamente as mais curtas (ROODBERGEN; KOSTER, 2001a).

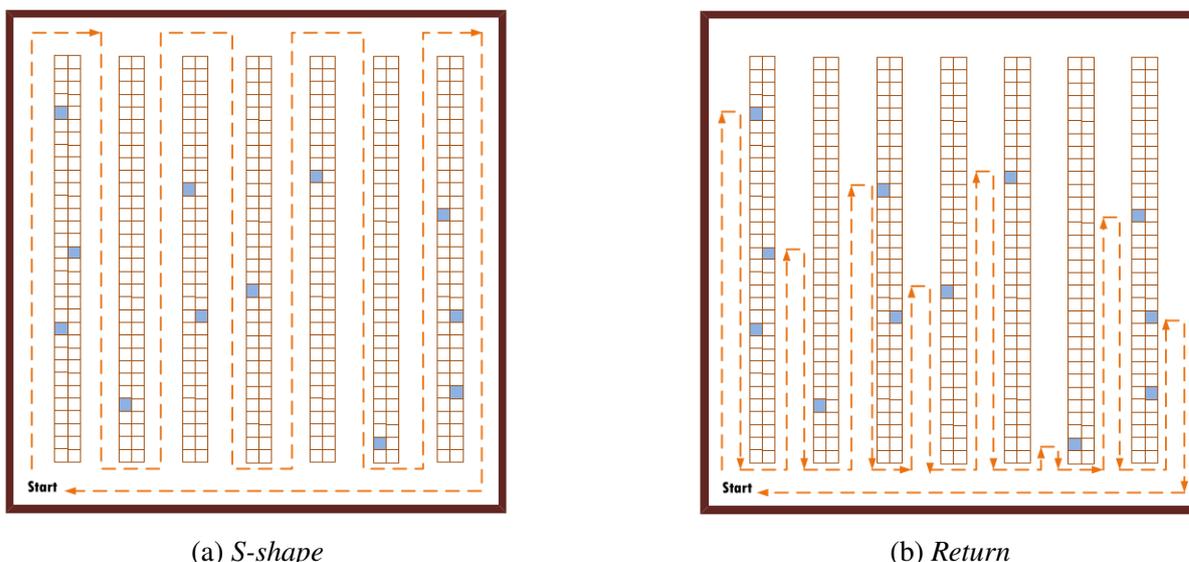
Uma das justificativas para implementar as técnicas clássicas de roteamento é que elas estabelecem diretrizes simples para os operadores seguirem, possivelmente resultando em uma redução de erros, mesmo que isso implique em um maior deslocamento (CHARLES, 1997; ROODBERGEN; KOSTER, 2001b). Entretanto, o desafio dessas abordagens clássicas de roteamento é a sua falta de adaptação a todos os tipos de *layout*, especialmente em casos com mais de dois corredores transversais (ROODBERGEN; KOSTER, 2001b).

2.1.2.2 Técnicas Clássicas de Roteamento

Em relação às estratégias clássicas de roteamento, também identificadas como heurísticas clássicas, destaca-se a utilização de alguns métodos ou políticas em especial: *S-shape*, *Return*, *Midpoint*, *Largest Gap* e *Composite* (ROODBERGEN; KOSTER, 2001a).

A política *S-shape* é uma das abordagens mais simples para a roteirização. Neste método, qualquer corredor que contenha pelo menos um item a ser coletado é percorrido, de modo que o corredor seja acessado por uma das extremidades e deixado pelo lado oposto. Após coletar

Figura 5 – Exemplo de políticas de rotas *S-shape* e *Return*

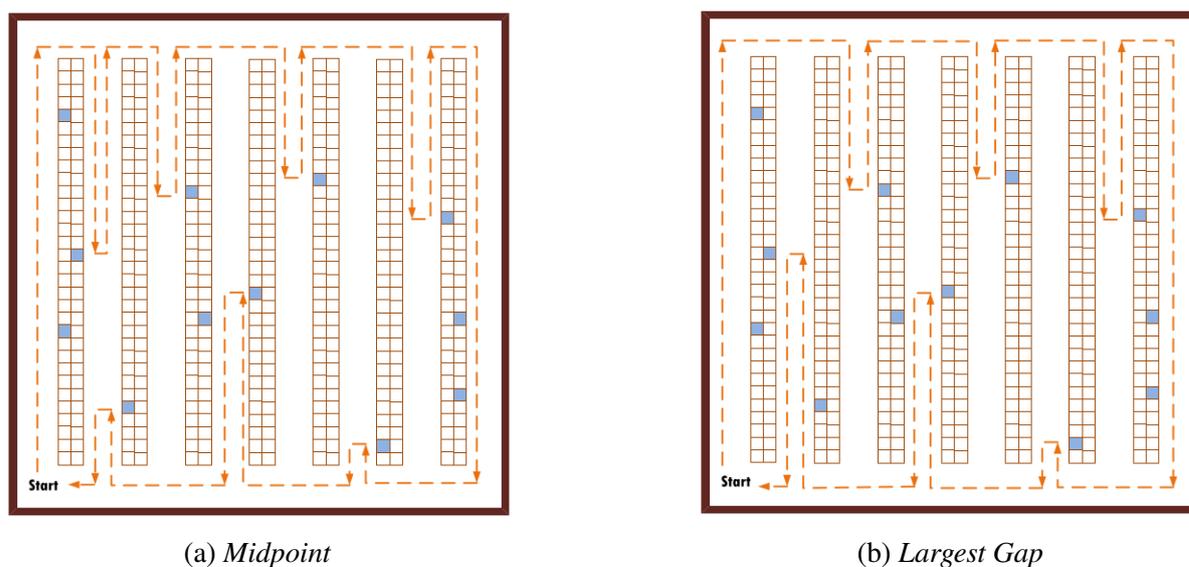


Fonte: Adaptado de Karasek (2013)

todos os itens, o operador retorna ao ponto de origem (ROODBERGEN; KOSTER, 2001a). Um exemplo desta política está apresentado na Figura 5a.

Na política *Return*, todos os corredores que contêm produtos a serem coletados são acessados e deixados pelo mesmo lado. Isso implica que um dos corredores transversais (frontal ou traseiro) não é percorrido durante uma rota específica de coleta (ROODBERGEN; KOSTER, 2001a). Um exemplo dessa rota é apresentado na Figura 5b

Figura 6 – Exemplo de políticas de rotas *Midpoint* e *Largest Gap*



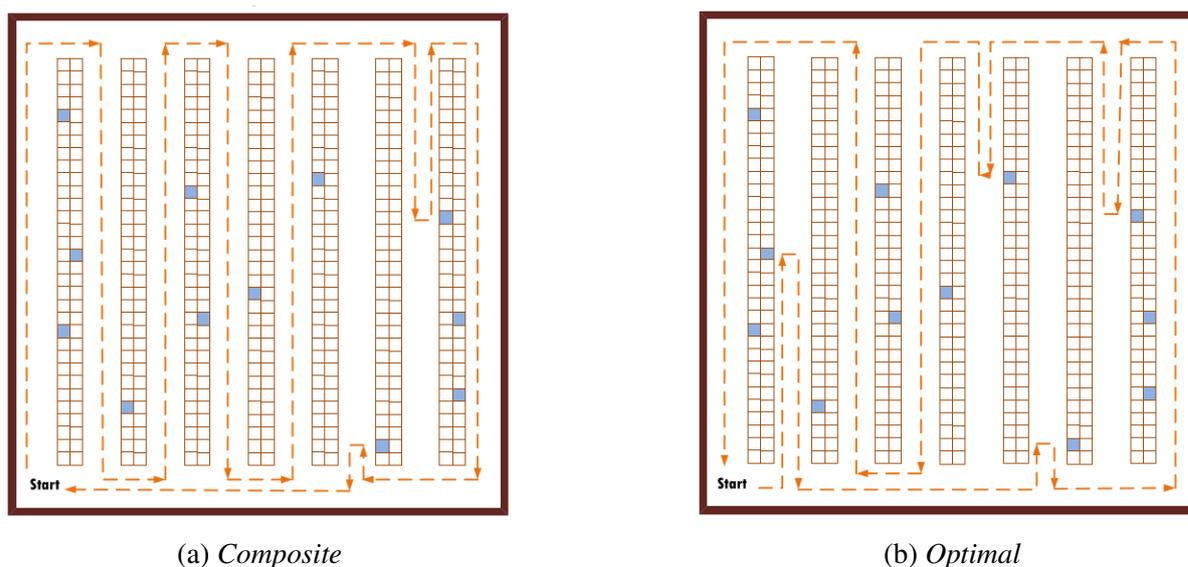
Fonte: Adaptado de Karasek (2013)

O método *Midpoint* (Figura 6a) divide o bloco ao meio, seguindo o eixo transversal ao comprimento das estantes. Nesta abordagem, o operador entra em um corredor que contém

pelo menos um item para coletar e percorre apenas até metade desse corredor, retornando pelo mesmo caminho que entrou. Uma vez que todos os itens de um lado do ponto médio tenham sido coletados, o operador se direciona para a metade oposta do corredor para coletar itens da mesma maneira, agora seguindo em direção ao ponto inicial (RENSBURG, 2019).

A política *Largest Gap* (Figura 6b) é semelhante ao método *Midpoint* no sentido de que o operador primeiro atravessa um corredor transversal na direção oposta ao ponto inicial, entrando e saindo do corredor pelo mesmo lado e depois retorna em direção ao ponto inicial ao longo do corredor transversal oposto. A diferença é que no método *Largest Gap*, os corredores são segmentados em intervalos de distância. Esses intervalos são calculados entre a distância do corredor frontal e o primeiro item a ser coletado, entre dois endereços adjacentes de coleta e entre o último item e o corredor traseiro. A maior dessas lacunas delimita uma seção do corredor que não será percorrida pelo operador, orientando-o a realizar a coleta pelo lado com o menor intervalo de distância (RENSBURG, 2019).

Figura 7 – Exemplo de políticas de rotas *Composite* e *Optimal*



Fonte: Adaptado de Karasek (2013)

A política *Composite* (Figura 7a) une as estratégias *Return* e *S-shape*. Em um corredor onde é necessário coletar um produto, a política *Return* é empregada quando a distância entre dois pontos pode ser reduzida em comparação com a distância resultante da aplicação da política do *S-shape*. Se a distância não pode ser minimizada, a rota segue a estratégia do *S-shape* (CHARLES, 1997).

Por fim, a política *Optimal* (Figura 7b) refere-se às rotas planejadas e calculadas de forma ideal para os operadores, ou seja, rotas que possuem a menor distância possível. Para encontrar rotas ótimas, resolve-se o PCV, empregando abordagens exatas ou heurísticas, como será explorado em Seções posteriores (CHARLES, 1997).

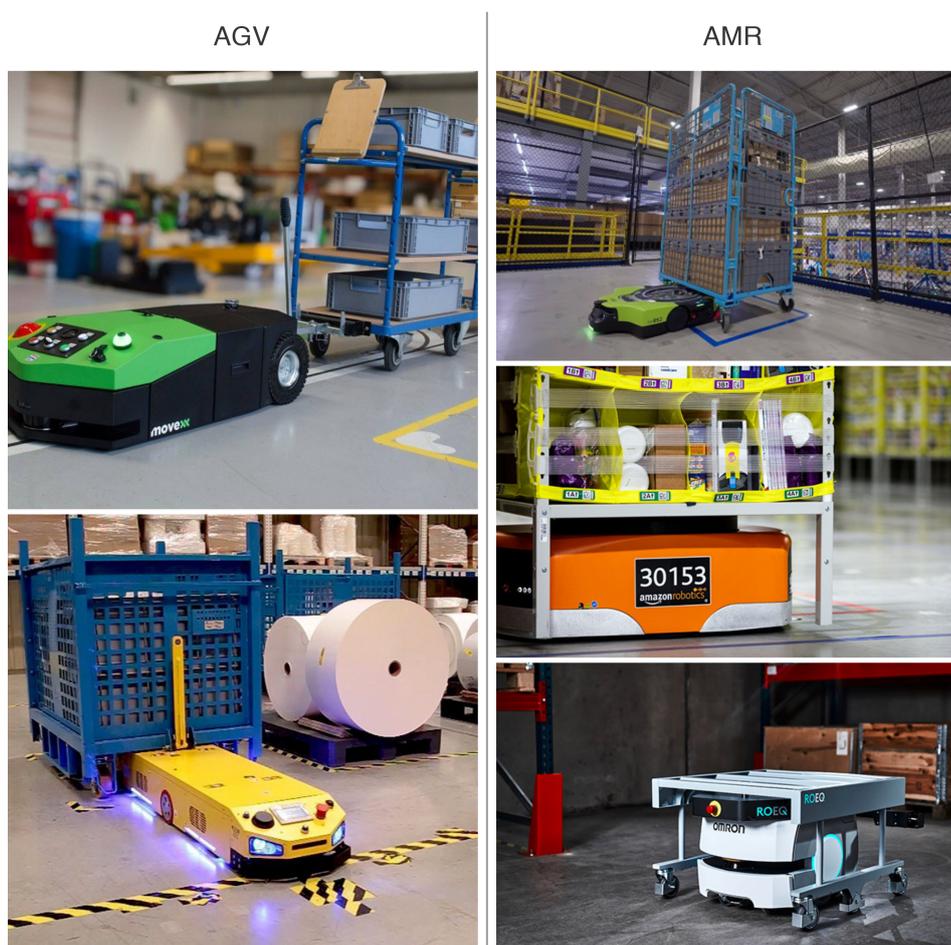
Para encontrar rotas subótimas, os pesquisadores têm proposto métodos meta-heurísticos,

como *Simulated Annealing* (GROSSE; GLOCK; BALLESTER-RIPOLL, 2014), Algoritmos Genéticos (ZHU, 2015) e Otimização por Colônia de Formigas (SANTIS et al., 2018).

2.2 TECNOLOGIAS DE MOVIMENTAÇÃO AUTÔNOMA

Nos últimos anos, houve um expressivo avanço nas tecnologias de movimentação e manipulação de materiais. Uma das notáveis evoluções é a transição dos AGVs para os AMRs. Os AGVs são os pioneiros na área, introduzidos na década de 1950 (MULLER, 1983), revolucionaram as operações logísticas ao proporcionar uma forma automatizada para transporte de carga em rotas predefinidas (GROOVER, 2008). Por outro lado, os AMRs, introduzidos conceitualmente em 1987 (MATTABONI, 1987), representam uma evolução tecnológica considerável como recursos de movimentação, oferecendo flexibilidade e autonomia para navegar em ambientes dinâmicos, de modo a adaptar-se aos imprevistos e às constantes mudanças de cenários em que estão inseridos (FRAGAPANE et al., 2021). A Figura 8 ilustra os AGVs percorrendo trajetos predefinidos, guiados pelas marcações presentes no piso, e os AMRs realizando navegação livre.

Figura 8 – Exemplos de AGVs e AMRs em diferentes operações

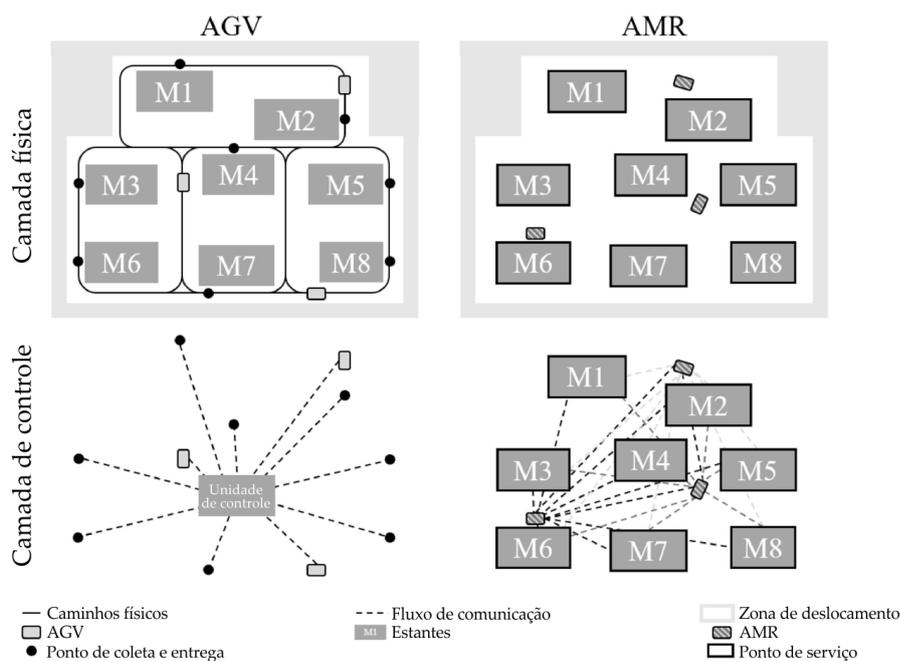


Fonte: Fotos da Web (2024)

Devido ao avanço das tecnologias de *hardware* e *software* observadas desde o surgi-

mento da tecnologia de movimentação baseada em AGVs, conforme descrito em detalhes no Apêndice C, os AMRs também se destacaram pela sua capacidade em realizar tomadas de decisão descentralizadas, diferente dos AGVs que dependem de uma unidade central. Os AMRs são equipados com recursos para se comunicar e negociar de forma independente com diversos sistemas e máquinas (FRAGAPANE et al., 2021).

Figura 9 – Contraste entre controles centralizado e descentralizado dos AGVs e AMRs



Fonte: Adaptado de Fragapane et al. (2021)

Ao analisar a Figura 9, observa-se claramente a distinção no controle de cada uma das tecnologias e em algumas características relacionadas aos seus sistemas de navegação, que comprovam as diferenças entre os dois equipamentos. No caso dos AGVs, observa-se a presença de caminhos físicos predefinidos que direcionam os veículos, acompanhados de uma unidade de controle centralizada responsável pela comunicação com todos os equipamentos envolvidos, a qual delega as ações a serem executadas por cada veículo da frota. Em contraste, nos AMRs, nota-se uma maior flexibilidade de movimentação, sem a necessidade de seguir caminhos fixos, permitindo-lhes transitar em qualquer direção e local. Além disso, o controle descentralizado dos AMRs possibilita que cada robô se comunique tanto entre si quanto com os outros sistemas. Essa interação possibilita que cada robô compreenda seu ambiente operacional em tempo real e faça escolhas autônomas para cumprir seus objetivos de maneira eficiente.

Com o resultado de uma maior flexibilidade e autonomia, os AMRs são capazes de operar em uma variedade maior de ambientes do que os AGVs, incluindo armazéns, fábricas, *cross-docks*, terminais de carga e hospitais (FRAGAPANE et al., 2021). Além disso, estas características permitem aos AMRs a possibilidade de auxiliar, colaborar e interagir de forma segura e eficiente com humanos e outras máquinas, ampliando seu escopo de aplicação. Um exemplo

específico de colaboração entre humanos e robôs, que se alinha ao foco deste estudo, será explorado em detalhes na Seção seguinte.

2.3 COLABORAÇÃO HUMANO-ROBÔ NA SEPARAÇÃO DE PEDIDOS

Conforme destacado por Azadeh, Koster e Roy (2019), a pesquisa acerca da colaboração entre humanos e robôs em armazéns começou a se desenvolver apenas recentemente.

No processo colaborativo entre humanos e robôs observado na operação de separação de pedidos, os operadores humanos trabalham em conjunto com um AGV ou AMR, coletando itens dos locais de armazenamento de acordo com o pedido recebido. O operador humano é responsável por coletar manualmente os itens dos endereços de armazenagem e colocá-los nas estruturas de armazenamento que estão equipadas nos robôs (Figura 10), que retornam autonomamente ao depósito assim que o pedido é concluído, permitindo que o operador permaneça na área de armazenagem, enquanto um novo AGV ou AMR é despachado para se juntar ao operador para iniciar a coleta do próximo pedido (LÖFFLER; BOYSEN; SCHNEIDER, 2021).

Figura 10 – Coleta de pedidos realizada com auxílio de um robô



Fonte: SSI SCHAEFER (2020)

Segundo Löffler, Boysen e Schneider (2021), dentre os principais benefícios acerca da implementação deste conceito destaca-se a diminuição dos deslocamentos dos operadores, permitindo a coleta contínua de pedidos sem retornos frequentes ao depósito, o que aumenta a eficiência por turno e reduz a inatividade. A implementação do uso de AMRs no processo de coleta de pedidos é simples e exige pouco esforço organizacional em sistemas de separação já existentes, além de ser escalável para atender demandas variáveis, como picos sazonais. Porém, a implementação desses sistemas exige maiores investimentos e enfrenta desafios na determinação de como empregar o recurso de automação de forma otimizada.

No estudo de Žulj et al. (2022), é apresentado um exemplo prático aplicado à política de coleta por zona (Seção 2.1.2), na qual cada operador humano é designado a um corredor

específico no armazém. Nesse estudo, os operadores coletam manualmente os itens de suas respectivas zonas e, em seguida, carregam esses itens em um AMR que se direcionará a um endereço predefinido, no final de cada corredor. Após visitar todos os corredores e coletar os itens, o AMR é responsável pelo transporte dos produtos da área de armazenamento para o depósito.

2.3.1 Problemas de decisão associados à adoção desta tecnologia

De acordo com Boysen, de Koster e Weidinger (2019), uma das questões envolvidas na implementação de sistemas de coleta colaborativa humano-robô é determinar o tamanho ideal da frota de veículos autônomos. Uma frota pequena seria ideal para economizar em custos de aquisição e manutenção, mas se for insuficiente, de modo que resulte em esperas frequentes dos operadores pelos robôs, não irá promover melhoria real no desempenho da operação de coleta. Portanto, os autores sugerem no mínimo um veículo por operador, embora o ideal seriam dois veículos para cada operador, dado que isso permite que um AGV ou AMR complete a entrega dos itens recém coletados enquanto o outro já esteja disponível para o operador dar sequência ao processo de coleta seguinte.

Além do dimensionamento de frotas, outros aspectos fazem parte dos principais problemas de decisão que acompanham a implementação de sistemas desta natureza, conforme discutido anteriormente. Estudos são demandados para uma utilização eficiente do sistema de separação de pedidos assistido por robôs autônomos, como por exemplo:

- a) Estratégias de Armazenamentos (Apêndice A): Os autores Löffler, Boysen e Schneider (2021) destacam que aplicar as políticas tradicionais, como, por exemplo, armazenar SKUs de alta demanda próximas do depósito, pode não ser a melhor opção. No processo colaborativo, onde os operadores permanecem na área de armazenamento/separação, alocar esses produtos no centro do armazém pode ser uma alternativa mais eficiente;
- b) Estratégias de separação (Seção 2.1.2): Conforme Boysen, de Koster e Weidinger (2019), é recomendável adotar um agrupamento de pedidos (*Batch Picking*) em situações onde os pedidos dos clientes são pequenos e contêm poucos itens. Agrupar estes itens em rotas de separação maiores, de modo que possam ser colocados em recipientes separados no AGV ou AMR eliminando a etapa de consolidação final, pode significativamente aumentar a eficiência;
- c) Rotas de coleta (Seção 2.1.2.1): Como mencionado anteriormente, o problema de roteamento do coletor deve determinar a ordem dos endereços a serem visitados durante a separação de produtos, com o objetivo de minimizar a distância percorrida ou o tempo em deslocamento. Nos processos convencionais sem o suporte de AGVs ou AMRs, o trajeto do operador poderia ser otimizado, já que ele inicia e conclui a separação dos pedidos em um depósito central. Isso resulta na decomposição de problemas individuais para cada

pedido. Por outro lado, na coleta com suporte de AGVs ou AMRs, o operador executa uma sequência de coletas de pedidos sem a necessidade de retornar ao depósito central. Portanto, este problema não se fragmenta em problemas independentes, mas sim, em um único problema que requer a determinação de uma sequência ótima de pedidos que proporcione determinar as rotas mais eficientes levando em consideração todos os pedidos (FRAGAPANE et al., 2021).

O presente estudo busca contribuir com o terceiro problema, isto é, a construção de rotas de coleta eficientes e alinhadas à operação de coleta na forma colaborativa humano-robô.

2.4 OTIMIZAÇÃO E MODELAGEM

Esta seção apresenta os fundamentos da otimização e modelagem, com ênfase nos problemas clássicos de roteamento e nas técnicas utilizadas para resolvê-los. São abordados métodos exatos e heurísticos aplicados à solução de problemas combinatórios, com foco no Problema do Caixeiro Viajante e suas variantes.

2.4.1 Problemas de Otimização: Fundamentos e Modelagem

Um problema de otimização tem como objetivo encontrar a melhor solução a partir de um conjunto de possíveis soluções dentro de um domínio específico. Ele pode ser formalizado como $P = (S, f, \Omega)$, onde S representa o espaço de busca definido pelos possíveis valores de um conjunto de variáveis x_1, x_2, \dots, x_n , chamadas de variáveis de decisão, f é a Função Objetivo (FO) e Ω denota o conjunto de restrições do problema. A função objetivo avalia a qualidade das soluções e orienta o processo de busca, enquanto as restrições regulam a viabilidade das soluções dentro do domínio do problema. O objetivo é encontrar a solução ótima que maximize ou minimize a função objetivo enquanto satisfaz as restrições (COOK et al., 1998; KORTE; VYGEN, 2012).

Os problemas de otimização podem ser divididos em: problemas de otimização contínua, que envolvem variáveis de decisão definidas em um domínio contínuo, onde tipicamente existe um número infinito de soluções válidas e, problemas de otimização combinatória, que envolvem variáveis de decisão definidas em um domínio discreto e que apresentam um número finito de soluções. Alguns métodos de busca pela solução de problemas de otimização contínua podem usar informações de derivadas e gradientes para descrever a inclinação de uma função, diferentemente das estratégias de solução de problemas de otimização combinatória, que não têm essa possibilidade, tornando sua resolução mais complexa (RABINER, 1984). Na Seção 2.5, são exploradas algumas estratégias para abordar os problemas combinatórios. Este trabalho focará especificamente nessa classe de problemas, a qual constitui a base dos desafios que este estudo se propõe a resolver.

Na representação de problemas de otimização, comumente os modelos utilizam estruturas matemáticas que abstraem e representam a realidade de forma matemática. Embora a modelagem matemática possa conter algumas simplificações, seu objetivo é representar a realidade da forma mais fiel possível. Para isso, alguns elementos precisam ser definidos, sendo eles: os parâmetros do modelo, que consistem em um conjunto de coeficientes constantes utilizados tanto na função objetivo quanto nas restrições; as variáveis de decisão, representando o número n de escolhas que necessitam ser efetuadas; a função objetivo, que é uma expressão matemática denotando a métrica para avaliar os benefícios (maximização) ou desvantagens (minimização) das decisões tomadas; e as restrições, que são um conjunto de limitações aplicadas às variáveis de decisão. A formulação genérica de um modelo de otimização pode ser assim representado:

$$\min \text{ ou } \max f(x_1, x_2, \dots, x_n) \quad (2.1)$$

sujeito a:

$$h_k(x_1, x_2, \dots, x_n) = a_k, \quad k = 1, 2, \dots, l; \quad (2.2)$$

$$g_i(x_1, x_2, \dots, x_n) \leq b_i, \quad i = 1, 2, \dots, m; \quad (2.3)$$

$$x_j \in S_j, \quad j = 1, 2, \dots, n. \quad (2.4)$$

onde:

- f é a função objetivo;
- x_1, x_2, \dots, x_n são as variáveis de decisão;
- h_k são as funções que representam as restrições de igualdade;
- g_i são as funções que representam as restrições de desigualdade;
- a_k são os limites para as restrições de igualdade;
- b_i são os limites para as restrições de desigualdade;
- S_j é conjunto de valores possíveis para a variável x_j ;
- m é o número de restrições;
- l é o número de restrições de desigualdade;
- n é o número de variáveis de decisão.

Uma solução é denominada viável se um vetor $\mathbf{x} = (x_1, \dots, x_n)$ cumpre as condições (2.2) a (2.4), indicando que ela satisfaz todas as restrições estabelecidas pelo problema e, portanto, pertence ao conjunto de soluções viáveis S_i . Um modelo de otimização que possui ao menos uma solução viável é considerado factível.

A depender da forma matemática assumida pela função objetivo f e as restrições h e g , diferentes modelos de otimização podem ser identificados (GOLDBARG; LUNA, 2005). Se

alguma destas funções assume a forma não-linear, tem-se um modelo de programação não-linear. Se todas estas funções assumem a forma linear, tem-se um modelo de programação linear. Por sua vez, os modelos de programação linear podem ser categorizados em quatro tipos distintos: i) Programação Linear, quando todas as variáveis x_i são contínuas; ii) Programação Linear Inteira, quando todas as suas variáveis x_i são restritas a valores inteiros; iii) Programação Linear Binária, na qual as variáveis x_i estão limitadas a assumir apenas os valores 0 ou 1; e iv) Programação Linear Inteira Mista, que se distingue por restringir somente uma parte das variáveis x_i a valores inteiros, permitindo que as demais variáveis assumam valores contínuos.

2.4.2 Modelos clássicos de otimização de rotas

Esta Seção descreve brevemente alguns modelos de problemas clássicos de otimização combinatória, com aplicação direta na solução de problemas de roteamento. Por meio desta descrição, busca-se uma compreensão detalhada do Problema do Caixeiro Viajante (PCV) e do Problema do Caixeiro Viajante Aberto (PCVA), que são fundamentais na composição da metodologia utilizada neste estudo.

Os problemas de roteamento envolvem encontrar as rotas mais eficientes para a distribuição de recursos ou informações, minimizando distâncias, tempo ou outra medida de custo, considerando restrições como, por exemplo, capacidade e janelas de tempo. Alguns exemplos incluem o problema do carteiro chinês (*Chinese Postman Problem*) (GUAN, 1962), problema de roteamento de veículos (*Vehicle Routing Problem*) (DANTZIG; RAMSER, 1959), além do Problema do Caixeiro Viajante (PCV) (DANTZIG; FULKERSON; JOHNSON, 1954) e sua variante, o Problema do Caixeiro Viajante Aberto (PCVA) (CHRISTOFIDES, 1970; BOFFEY, 1982), que serão explorados em detalhes nas Seções 2.4.2.1 e 2.4.2.2, respectivamente.

2.4.2.1 Problema do Caixeiro Viajante (PCV)

O Problema do Caixeiro Viajante (*Traveling Salesman Problem - TSP*) é um problema clássico em otimização combinatória e ciência da computação. No PCV, um caixeiro viajante recebe uma lista de cidades e os custos ou distâncias entre elas. O objetivo é encontrar a rota de menor custo possível que visita cada cidade exatamente uma vez e retorna à cidade de origem (VOUDOURIS; TSANG, 1999). Esse problema, em sua forma original, consiste em encontrar o ciclo Hamiltoniano de menor custo em um grafo ponderado.

Formalmente, o PCV pode ser definido considerando um grafo completo $G = (V, A)$, onde V é o conjunto de n vértices e A é o conjunto de arestas ou arcos que conectam cada par de cidades i e $j \in V$. Cada aresta está associada a um custo $c_{i,j}$. O PCV consiste em encontrar a rota de menor custo, passando por cada vértice uma única vez. No caso simétrico, $c_{i,j} = c_{j,i}$ para toda cidade $i, j \in V$, já no caso assimétrico existe ao menos um caso em que $c_{i,j} \neq c_{j,i}$.

Sabe-se que o PCV é um problema NP-difícil, o que significa que não se conhece um algoritmo de tempo polinomial que possa resolver todas as instâncias do problema de forma

ótima (PAPADIMITRIOU, 1977). Por exemplo, considere um conjunto de n cidades, as rotas podem ser representadas por permutações dos números de 1 a n , formando ciclos que conectam as cidades na ordem da permutação (indicando a sequência a ser seguida) e retornam à cidade inicial. O conjunto de todas as permutações constitui o espaço das soluções possíveis, portanto o número total de escolhas é dado pela notação fatorial $n!$. Ou seja, para calcular o número total de rotas para n cidades, pode-se pensar da seguinte forma:

1. Qualquer uma das n cidades pode ocupar a primeira posição na permutação;
2. Na segunda posição, pode-se escolher qualquer uma das $n - 1$ cidades restantes;
3. Analogamente, na terceira posição, tem-se $n - 2$ opções, e assim por diante.

Dessa maneira, o número total de escolhas possíveis é dado por $n \times (n - 1) \times (n - 2) \times \dots \times 2 \times 1$, equivalente a $n!$. Por outro lado, se a distância entre duas cidades for simétrica, o número de diferentes combinações será reduzido pela metade, sendo expresso por $n!/2$. Por exemplo, para resolver o PCV para um caso hipotético de apenas 22 cidades com distâncias assimétricas entre si, o número de possíveis rotas seria $22!$, ou seja, $5,1091 \times 10^{19}$ diferentes possibilidades. Esse raciocínio mostra que resolver problemas caracterizados pelo fenômeno da explosão combinatória apresentam uma dificuldade associada ao crescimento exponencial, da quantidade de soluções possíveis, que é determinado de acordo com o tamanho da instância a ser resolvida.

Modelagem

Neste estudo, adota-se a formulação de Miller-Tucker-Zemlin (MTZ), proposta originalmente para solucionar o problema do caixeiro viajante, com o objetivo de prevenir a formação de subrotas na solução, conforme Miller, Tucker e Zemlin (1960). A seguir, apresenta-se a sua formulação matemática, incluindo, parâmetros, variáveis de decisão e as equações do modelo. Visando a facilitar o entendimento, os significados dos símbolos utilizados na modelagem são ajustados ao contexto abordado neste estudo.

Parâmetros

- n : número de pontos de coleta a serem visitados no roteiro de separação;
- c_{ij} : custo para percorrer o trecho do ponto de coleta i até o ponto de coleta j .

Variáveis de decisão

- x_{ij} : igual a 1 se o robô percorre o trecho (i, j) , 0 caso contrário;
- u_i : variável contínua associada ao ponto de coleta i .

Modelo matemático

$$\min \sum_{i=1}^n \sum_{j=1, j \neq i}^n c_{ij} x_{ij} \quad (2.5)$$

sujeito a:

$$\sum_{i=1, i \neq j}^n x_{ij} = 1, \quad j = 1, 2, \dots, n; \quad (2.6)$$

$$\sum_{j=1, j \neq i}^n x_{ij} = 1, \quad i = 1, 2, \dots, n; \quad (2.7)$$

$$u_i - u_j + n x_{ij} \leq n - 1, \quad 2 \leq i \neq j \leq n; \quad (2.8)$$

$$x_{ij} \in \{0, 1\}, \quad i, j = 1, 2, \dots, n \text{ para } i \neq j; \quad (2.9)$$

$$u_i \in \mathbb{R}^+, \quad i = 1, 2, \dots, n. \quad (2.10)$$

No modelo, a função objetivo (2.5) representa o custo total da rota do robô a ser minimizado. O custo total é expresso pelo produto entre a variável c_{ij} e a variável de decisão binária x_{ij} que indica se o robô se desloca do ponto de coleta i para o ponto de coleta j . A restrição (2.6) garante que cada ponto de coleta i seja origem de um único destino. A restrição (2.7) garante que o robô chegue a cada ponto de coleta j a partir de uma única origem, ou seja, as duas primeiras restrições garantem que cada ponto de coleta seja visitado uma única vez, garantindo que, caso seja ponto de chegada de um trecho, seja também ponto de partida do próximo trecho. A restrição (2.8) impede a criação de sub-rotas, forçando a ordem em que os pontos de coleta são visitados. A restrição (2.9) restringe a variável de decisão x_{ij} a ser binária, ou seja, assumam apenas o valor 0 ou 1. A restrição (2.10) indica que a variável de decisão u_i pertence ao conjunto dos números reais positivos.

2.4.2.2 Problema do Caixeiro Viajante Aberto (PCVA)

O Problema do Caixeiro Viajante Aberto (*Open TSP* ou *OTSP*) é uma variante do PCV clássico em que o caixeiro viajante não precisa necessariamente retornar à cidade de origem. Portanto, a formulação do PCVA é semelhante àquela do PCV, com a distinção de que não inclui a restrição de que o caixeiro viajante deve concluir sua rota na cidade de origem.

Modelagem

Para modelar o PCVA, adapta-se a formulação de Miller-Tucker-Zemlin previamente descrita (ver Seção 2.4.2.1), removendo a restrição que exige o retorno do caixeiro à cidade de origem ao concluir sua rota (CIČKOVÁ; BREZINA; PEKÁR, 2013). A formulação matemática do Problema do Caixeiro Viajante Aberto aplicada ao contexto deste estudo é apresentada a seguir, incluindo parâmetros, variáveis de decisão e as equações que constituem o modelo.

Parâmetros

- n : número de pontos de coleta a serem visitados no roteiro de separação;
- d_{ij} : custo para percorrer o trecho do ponto de coleta i até o ponto de coleta j .

Variáveis de decisão

- x_{ij} : igual a 1 se o robô percorre o trecho (i, j) , 0 caso contrário;
- u_i : variável contínua associada ao ponto de coleta i .

Modelo matemático

$$\min \sum_{i=1}^n \sum_{j=1}^n d_{ij} x_{ij} \quad (2.11)$$

sujeito a:

$$\sum_{i=1, i \neq j}^n x_{ij} = 1, \quad j = 2, 3, \dots, n; \quad (2.12)$$

$$\sum_{j=2, j \neq i}^n x_{ij} \leq 1, \quad i = 1, 2, \dots, n; \quad (2.13)$$

$$\sum_{j=2}^n x_{1j} = 1, \quad i = 2, 3, \dots, n; \quad (2.14)$$

$$u_i - u_j + n x_{ij} \leq n - 1, \quad i, j = 2, 3, \dots, n \text{ para } i \neq j; \quad (2.15)$$

$$x_{ij} \in \{0, 1\}, \quad i, j = 1, 2, \dots, n \text{ para } i \neq j. \quad (2.16)$$

No modelo, a função objetivo (2.11) representa o custo total da rota a ser minimizado, sendo representado pelo produto entre a variável d_{ij} e uma variável de decisão binária x_{ij} que indica se o robô se desloca do ponto de coleta i para o ponto de coleta j . A restrição (2.12)

garante que cada ponto de coleta deve ser visitado uma única vez, o que significa que a variável x_{ij} é igual a 1 para exatamente um ponto de coleta j (que não seja i). A restrição (2.13) assegura que o robô não precisa sair de cada ponto de coleta, uma vez que a rota pode se encerrar após a visita ao último ponto de coleta, ou seja, para cada ponto de coleta i (exceto o de origem), há no máximo um trecho saindo deste ponto de coleta. A restrição (2.14) assegura que, para cada ponto de coleta i (diferente do inicial 1), haja exatamente um trecho ligando a origem a esse ponto de coleta i na rota, ou seja, não precisa retornar ao início da rota. A restrição (2.15) impede a criação de sub-rotas, forçando a ordem em que os pontos de coleta são visitados. A restrição (2.16) especifica que as variáveis de decisão x_{ij} só podem assumir valores binários, isto é, valor 0 ou 1. Ao contrário do modelo para o PCV, no caso do modelo para o PCVA, não há uma restrição explícita que obrigue o robô retornar ao ponto de origem da rota de separação. Isso permite que o percurso realizado seja “aberto”, proporcionando uma maior flexibilidade no planejamento das rotas, uma vez que o modelo permite que o ponto de encerramento da viagem, após visitar todos os outros pontos de coleta, possa ser escolhido livremente.

A partir das formulações matemáticas do PCV e do PCVA, fica evidente a complexidade envolvida na resolução desses problemas, caracterizada principalmente pela explosão combinatória. Devido a esse crescimento exponencial do espaço de soluções, torna-se essencial utilizar técnicas eficientes para obter soluções viáveis em tempo hábil. Nesse contexto, a próxima seção aborda estratégias para resolver problemas de otimização combinatória, incluindo métodos exatos, que garantem soluções ótimas, e métodos aproximados (heurísticas e meta-heurísticas), que fornecem soluções rápidas e aceitáveis para aplicações práticas.

2.5 ESTRATÉGIAS DE SOLUÇÃO NA OTIMIZAÇÃO COMBINATÓRIA

A otimização combinatória é focada na resolução de problemas que exigem a seleção de um conjunto específico e finito de valores de um universo de soluções possíveis, atendendo a critérios predefinidos. Para isso, Wolsey e Nemhauser (2014) explicam que a busca pela solução desta classe de problemas é guiada por uma função objetivo e sujeita a restrições de igualdade e desigualdade, que definem as combinações possíveis dos valores das variáveis sem que as restrições sejam violadas, determinando assim, as soluções viáveis dentro do espaço de busca.

O foco desses problemas envolve a identificação de uma solução ótima que maximize ou minimize algum valor almejado. Essa busca pela solução ideal deve ocorrer dentro de um espaço delimitado por restrições impostas ou desejadas e pode considerar a escassez de recursos disponíveis, como máquinas, pessoas, veículos, matéria-prima, entre outros aspectos (KORTE; VYGEN, 2012).

Problemas de otimização combinatória apresentam diversas abordagens para sua resolução, variando de acordo com sua complexidade, relacionada ao espaço de soluções e restrições, mas também relacionada a outros fatores como tempo e recursos disponíveis. A complexidade destes problemas torna fundamental a seleção de algoritmos eficientes e escaláveis. Uma ma-

neira de avaliar a eficácia e a eficiência de um algoritmo consiste em analisar sua complexidade.

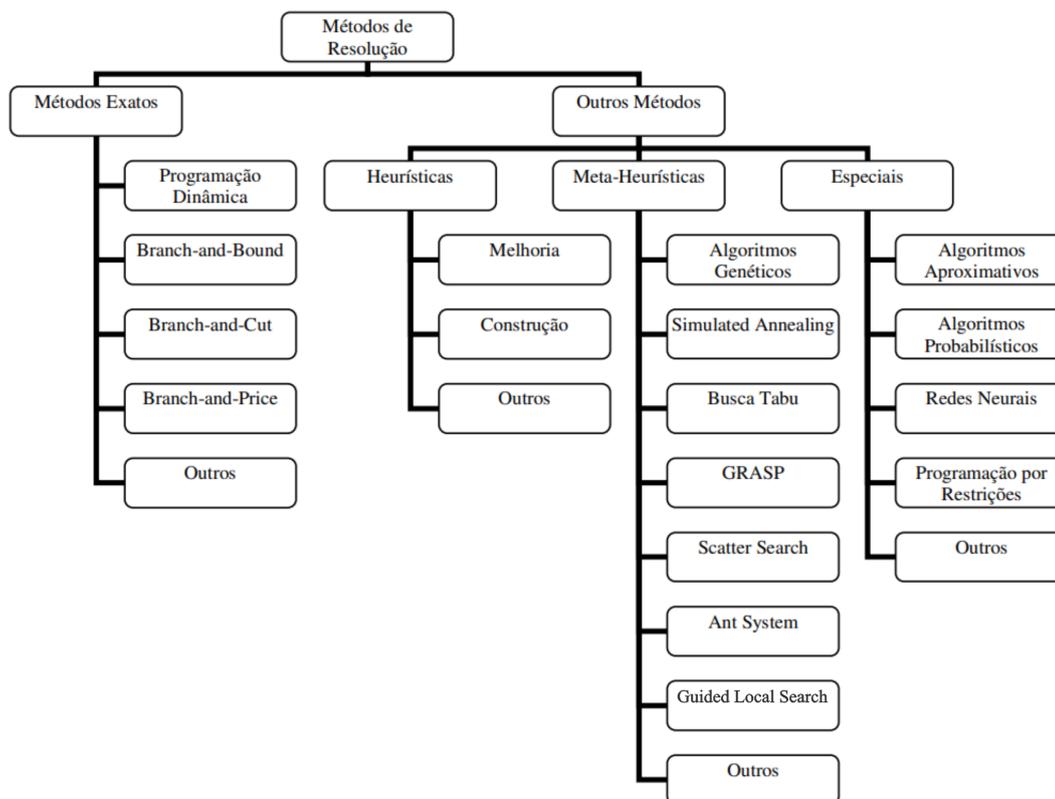
A complexidade dos algoritmos envolve uma análise que leva em consideração o tempo de execução em relação ao tamanho da entrada. Ela considera o número de operações básicas realizadas durante a execução do algoritmo e fornece uma indicação do tempo de execução como uma função do tamanho da entrada. É uma ferramenta comum para comparar o desempenho de diferentes algoritmos, pois não precisa levar em consideração as características do *hardware* em que são executados. Essa abordagem de avaliação, focada no crescimento do tempo de execução em relação ao tamanho da entrada, é essencial para compreender a eficiência assintótica dos algoritmos e distinguir se eles são problemas de decisão fáceis ou difíceis (CORMEN et al., 2009).

O conjunto de soluções possíveis é conhecido como espaço de busca e para um problema de otimização ele pode aumentar exponencialmente com base no tamanho da entrada. De fato, essa característica está presente na maioria dos problemas combinatórios encontrados em aplicações reais (RABINER, 1984). Desta forma, os problemas de otimização combinatória que são a classe de problemas que envolvem o presente estudo, frequentemente são classificados como NP-difíceis, portanto, não se conhecem métodos determinísticos capazes de resolvê-los em tempo polinomial (RABINER, 1984; PAPADIMITRIOU; STEIGLITZ, 1998). Isso significa que, o tempo necessário para fornecer a solução cresce de maneira exponencial de acordo com o tamanho dos dados de entrada.

No contexto de problemas NP-Difíceis, os métodos para resolvê-los se dividem em duas categorias que possuem diferentes focos. A primeira, dos métodos exatos, que buscam encontrar soluções comprovadamente ótimas, estabelecendo limites superiores e inferiores para acelerar a enumeração no espaço de busca. A segunda, das heurísticas, que têm como prioridade a eficiência operacional, de modo a explorar apenas subconjuntos de possíveis soluções promissoras na tentativa de encontrar um bom resultado. Esta abordagem diminui o consumo de recursos computacionais necessários para estabelecer os limites inferiores e superiores. Contudo, ela abre mão da garantia de identificar a solução ótima (WOLSEY; NEMHAUSER, 2014). A Figura 11 mostra a distribuição dos principais métodos de resolução de problemas de otimização entre as categorias destacadas anteriormente.

Tanto os métodos exatos quanto os heurísticos foram originalmente desenvolvidos para problemas específicos e depois evoluíram para se tornarem estruturas mais abrangentes, capazes de resolver problemas combinatórios gerais. Com o surgimento da Programação Linear Inteira Mista (PLIM) (WOLSEY; NEMHAUSER, 2014), métodos exatos importantes, como *simplex* (GOMORY, 1960), *cutting-plane* (DANTZIG; FULKERSON; JOHNSON, 1954) e *branch-and-bound* (LAWLER; WOOD, 1966), foram introduzidos. Posteriormente, surgiram estruturas heurísticas e meta-heurísticas genéricas, como por exemplo, *local search* (AARTS; AARTS; LENS-TRA, 2003), *Genetic Algorithm* (GA) (GOLDBERG; HOLLAND, 1988), *Simulated Annealing* (SA) (KIRKPATRICK; GELATT; VECCHI, 1983) e *Guided Local Search* (GLS) (VOU-

Figura 11 – Métodos de resolução de problemas de otimização



Fonte: Adaptado de Marcondes Filho (2008a)

DOURIS; TSANG, 1999).

Nas Seções a seguir, é fornecida uma visão geral dos métodos de solução relevantes para este estudo, empregados como abordagens para resolver o PCV e o PCVA.

2.5.1 Métodos Exatos

Métodos exatos são abordagens capazes de fornecerem uma solução com a garantia de que ela seja a solução ótima global. Isso significa que, dentro dos critérios e contexto estabelecidos para a otimização, esses métodos são capazes de confirmar que a solução encontrada é a melhor possível, garantindo que nenhuma outra solução existente possa oferecer um resultado superior. Modelos de Programação Linear (PL) estão associados a importantes técnicas de otimização para alcançar o ótimo global em situações com múltiplas escolhas e restrições (WOLSEY; NEMHAUSER, 2014).

A abordagem mais comum para aplicar um método exato consiste na enumeração de todas as possíveis soluções através de um processo de força bruta. Contudo, essa técnica é limitada devido à explosão combinatória, fenômeno característico de problemas combinatórios, conforme já destacado. Tal características torna essa abordagem útil apenas para problemas com instâncias pequenas. Para superar essa limitação, foram desenvolvidas técnicas de enumeração implícita, como o *branch-and-bound*, que explora o espaço de busca de forma mais eficiente,

permitindo encontrar a solução ótima sem analisar todas as possibilidades. Métodos determinísticos, como o algoritmo simplex e algoritmos de gradiente, utilizam conhecimento específico do problema para navegar de forma sistemática pelo espaço de busca. Eles são classificados como determinísticos, porque, dadas as mesmas condições iniciais (parâmetros e valores), sempre produzem o mesmo resultado, sem variação (HOFFMAN; RALPHS, 2013; PERES; CASTELLI, 2021).

2.5.1.1 *Branch-and-Bound*

O método *Branch-and-Bound* (B&B) é um algoritmo exato de enumeração implícita, que garante soluções ótimas sem avaliar todas as alternativas viáveis. Ele se baseia na criação de uma árvore de subproblemas originada do problema principal, seguindo a estratégia “dividir para conquistar” e foca na otimização de subconjuntos específicos ao invés do conjunto total (BRUCKER, 2007). O método é composto por duas operações: *branching* e *bounding*.

O *branching* consiste na subdivisão do problema principal em subproblemas menores, com o objetivo de simplificar a análise e descartar soluções inviáveis, sem prejudicar a integridade do conjunto de soluções (HOFFMAN; RALPHS, 2013).

Já o processo de *bounding* consiste em excluir soluções subótimas por meio de comparações com valores limitantes, que são classificados em duas categorias: superior e inferior. No contexto de um problema de minimização, o limite superior representa um valor viável já conhecido da função objetivo. Embora não seja necessariamente o valor ótimo, ele atua como uma referência para a avaliação de novas soluções encontradas. Desse modo, quaisquer soluções que apresentem valores acima deste limite são descartadas, por serem consideradas piores em comparação à melhor solução até então identificada (HOFFMAN; RALPHS, 2013). Paralelamente, o limite inferior representa uma estimativa da função objetivo, baseada na solução parcial atual. Por isso, em um problema de minimização, o limite inferior é sempre menor ou igual ao valor real da função objetivo, visto que seu cálculo origina-se de uma fração da solução, enquanto a função objetivo engloba a solução completa. Dessa forma, é possível eliminar soluções que tenham limites inferiores piores do que os limites superiores atualmente conhecidos, dado que, se o menor valor possível (limite inferior) de uma solução parcial já é maior do que a melhor solução encontrada (limite superior), essa solução parcial não pode gerar resultados melhores. Portanto, as ramificações dessas soluções parciais podem ser descartadas, tornando a busca pelo valor ótimo mais eficiente (HOFFMAN; RALPHS, 2013). Um maior detalhamento do funcionamento do método B&B, é apresentado no Apêndice D.

2.5.1.2 *Branch-and-Cut*

O método *Branch-and-Cut* (B&C) é uma abordagem que combina um algoritmo de *branch-and-bound* com o uso de planos de corte de Gomory (1958), utilizados para melhorar a eficiência do processo de busca pela solução ótima (Apêndice E). Essa combinação tem

como objetivo reduzir o tempo computacional, especialmente em instâncias mais complexas, ao integrar restrições válidas no problema relaxado, sem eliminar soluções viáveis.

Este método utiliza um algoritmo de planos de corte em cada subproblema da árvore de ramificação, o que ajuda a gerar um limite inferior, reduzindo, assim, o número de nós da árvore de busca. Além disso, são adicionados cortes controlados, que são baseados em critérios específicos do problema em questão. Isso torna possível encontrar a solução ótima sem a necessidade do conjunto completo de restrições originais (HOFFMAN; RALPHS, 2013).

O método *branch-and-cut* também utiliza técnicas de pré-processamento e análises algébricas para melhorar a relaxação linear, eliminando redundâncias e fixando valores de variáveis sem afetar o resultado. Além disso, heurísticas e meta-heurísticas podem ser aplicadas para otimizar o desempenho, estabelecendo limites e reduzindo o tempo de processamento, o que resulta na diminuição do número de nós a serem explorados na árvore de busca (HOFFMAN; RALPHS, 2013). Um detalhamento do funcionamento do método B&C é apresentado no Apêndice E.

Os algoritmos utilizados na aplicação do método B&C, geralmente incluem procedimentos automáticos de reformulação e pré-processamento, heurísticas que geram soluções inteiras viáveis e desigualdades válidas, além dos procedimentos de *branch*. Esses componentes são combinados e aplicados em um método de divisão e conquista, semelhante à abordagem de *branch-and-bound*. Além disso, os algoritmos podem ser interrompidos de forma antecipada, com a capacidade de fornecer uma solução viável junto com uma estimativa do quão próxima ela está da solução ótima. Essa abordagem híbrida é utilizada para resolução de problemas gerais de PLIM, sendo a abordagem mais utilizada por *softwares* solucionadores para resolver esses tipos de problemas (HOFFMAN; RALPHS, 2013).

2.5.2 Métodos aproximados

Heurísticas e meta-heurísticas são métodos aproximados conhecidos por oferecer soluções eficientes, muitas vezes se aproximando ou encontrando a solução ótima, para problemas combinatórios que, de forma exata, seriam praticamente insolúveis em um tempo viável (PAPADIMITRIOU; STEIGLITZ, 1998). Além disso, esses métodos geralmente exigem um esforço computacional menor em comparação com abordagens exatas.

2.5.2.1 Heurísticas

Conforme Pearl (1984), as heurísticas podem ser definidas como critérios, métodos ou princípios utilizados na tomada de decisão, visando a promover alternativas mais eficazes entre várias opções disponíveis, para atingir um determinado objetivo. De forma geral, elas funcionam em conjunto com estratégias de busca e avaliações sistemáticas que exploram diferentes configurações e resolvem problemas complexos, cujas soluções exatas seriam demandariam um tempo excessivo para a sua determinação. Devido à sua natureza de produzir resultados aproximados, também conhecidos como soluções ótimas locais, as heurísticas podem não necessariamente

retornar a solução ótima para o problema que está sendo avaliado. Portanto, em troca do ganho de velocidade, essas abordagens geralmente incluem incertezas e não garantem resultados determinísticos (PERES; CASTELLI, 2021).

Em algumas estratégias heurísticas que apresentam limitações de exploração do espaço de soluções no decorrer do seu processo de busca, podem ser incorporados elementos estocásticos, que introduzem abordagens probabilísticas. Os métodos estocásticos, por sua vez, não dependem de conhecimento prévio sobre o problema e utilizam uma abordagem aleatória, embora guiada por probabilidades, para explorar o espaço de soluções (BIANCHI et al., 2009). Devido à sua natureza probabilística, os resultados variam a cada iteração, tornando-se úteis em cenários onde os dados são imprevisíveis (JUAN et al., 2015).

Os métodos heurísticos podem ser categorizados com base em sua abordagem e funcionamento. Uma das formas comuns de dividir esses métodos é em três categorias principais: heurísticas construtivas, heurísticas de melhoramento e heurísticas híbridas (NORDIN; MALAYSIA; SERDANG, 2016).

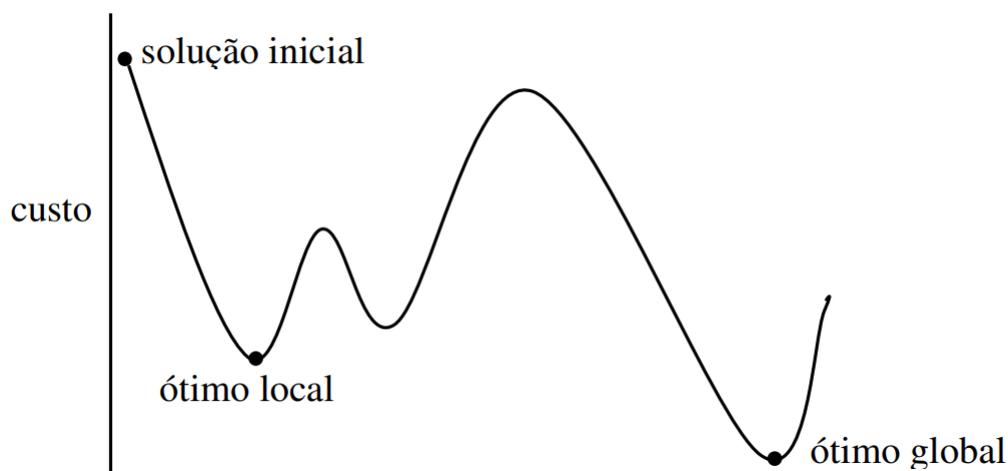
As heurísticas construtivas criam soluções de maneira incremental, analisando as variáveis de decisão e selecionando componentes um a um, podendo ou não avaliar a solução ao longo deste processo construtivo. De acordo com Silver (2004), esse método de construção gradual envolve a inclusão incremental de elementos à solução. A escolha de cada elemento geralmente está vinculada a uma função de avaliação específica para o problema em avaliação, como por exemplo, o vizinho mais próximo no caso do PCV. É importante ressaltar que essa função depende do tipo de problema que está sendo resolvido.

Heurísticas de melhoramento partem do princípio de que já existe uma solução, geralmente gerada por uma heurística construtiva, e são aplicadas para melhorar essa solução, realizando mudanças em busca de resultados melhores. Esse grupo de heurísticas é frequentemente utilizado em procedimentos de melhora iterativa denominados de busca local, que são técnicas que se concentram na exploração sistemática da vizinhança da solução atual (Seção 2.5.2.2). Essas abordagens funcionam com base em um critério de parada, que geralmente é acionado quando não se observam mais avanços na busca por uma melhor solução, alcançando o que é chamado de ótimo local. Entretanto, este ótimo local pode representar um ponto de estagnação, muitas vezes distante do ótimo global, que é a solução ideal do problema (SOUZA; FILHO, 2000). A Figura 12 ilustra essa situação.

Por fim, as heurísticas híbridas combinam elementos das abordagens construtiva e de melhoramento em um processo de busca local, buscando aproveitar as vantagens de ambas as técnicas, empregando uma estratégia de construção passo a passo da solução e seu aprimoramento iterativo, posteriormente (PERES; CASTELLI, 2021).

Para aplicação das heurísticas construtivas e de melhoramento, uma definição importante está relacionada à forma como a solução é representada, isto é, codificada computacionalmente.

Figura 12 – Ótimos local e global no espaço de busca em um problema de minimização



Fonte: Cordenosi (2008)

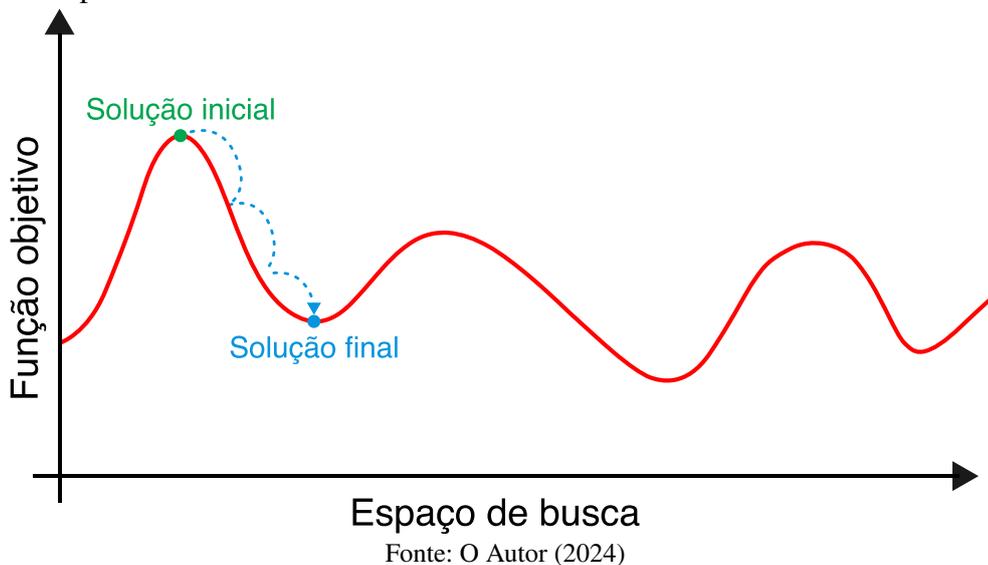
Para representar e codificar as soluções dos problemas de forma computacional, podem ser utilizados arranjos de elementos, tais como combinações ou permutações, para simbolizar as soluções. A organização das variáveis envolvidas é frequentemente realizada por meio de estruturas como sequências lineares (*arrays*), árvores e redes de nós interligados por arestas, que podem representar conexões unidirecionais ou bidirecionais (grafos). A seleção da estrutura apropriada desempenha um papel importante na definição das técnicas utilizadas para a manipulação e transformação das soluções, conforme discutido em (KORTE; VYGEN, 2012).

2.5.2.2 Processo de busca local

Um processo de Busca Local (*Local Search*) pode ser definido como um método que parte de uma solução inicial gerada por uma heurística construtiva ou algum outro procedimento que garanta a validade da solução fornecida (AARTS; AARTS; LENSTRA, 2003). Essa solução inicial serve de base para a geração de um conjunto de soluções próximas a ela, sendo também conhecidas como vizinhos ou vizinhança. As estruturas de vizinhança (heurísticas de melhoramento), por sua vez, são procedimentos que permitem alcançar o conjunto de soluções vizinhas a partir da solução atual fazendo movimentos específicos dentro do espaço de busca.

Por movimento em um espaço de busca, entende-se a aplicação de trocas, inversões, inclusões ou qualquer outra operação que permita modificar a solução atual, resultando na criação de uma nova solução. Isso provoca movimentos sequenciais que conduzem a solução desde o ponto inicial até alcançar o ponto da solução final, conforme ilustrado na Figura 13. Portanto, a busca local procura, iterativamente, aprimorar uma solução, explorando as estruturas de vizinhança em busca de melhores soluções, até que um critério de parada (como um número máximo de iterações ou um limite de tempo) seja alcançado, ou até que não haja mais melhorias significativas na solução atual (BRUCKER, 2007).

Figura 13 – Funcionamento de uma heurística de busca local aplicada a um espaço de busca específico



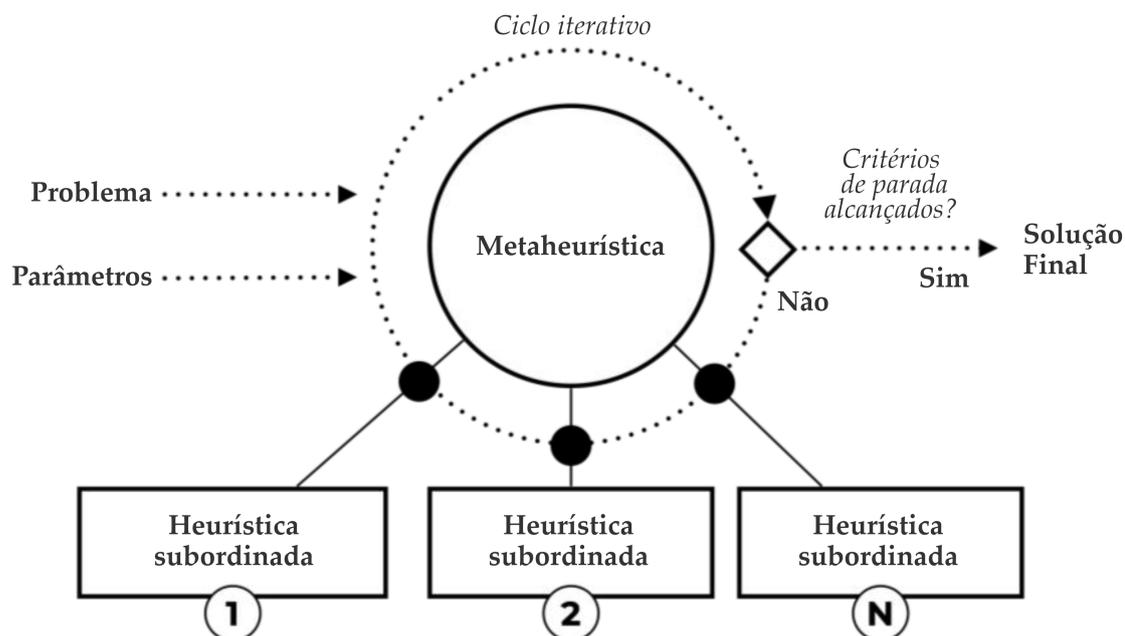
2.5.2.3 Meta-heurísticas

Para superar os problemas de estagnação impostos pelos ótimos locais observados no emprego das heurísticas de busca local, foram desenvolvidas as meta-heurísticas. Conforme descrito por Sörensen, Sevaux e Glover (2018), uma meta-heurística pode ser definida como uma abordagem de nível superior que consiste em uma heurística mestre responsável por orientar e coordenar um conjunto de heurísticas subordinadas em um ciclo iterativo, que tem como objetivo explorar de maneira eficiente o espaço de soluções sem ficar preso em ótimos locais. A interação entre a heurística mestre e suas heurísticas subordinadas pode ser observada na Figura 14.

As diferenças nos sistemas de navegação das meta-heurísticas para explorar o espaço de soluções se dividem em duas categorias: busca local (intensificação), que é focada em aprofundar a busca em uma região específica, e busca global (diversificação), que explora áreas mais amplas, mesmo que isso envolva aceitar movimentos para soluções piores durante o processo (HUSSAIN et al., 2019). O uso equilibrado das estratégias de intensificação e diversificação é fundamental para que uma meta-heurística tenha sucesso (BIANCHI et al., 2009; CREPINSEK; LIU; MERNIK, 2013).

No diagrama apresentado na Figura 14, o ciclo iterativo começa com as definições do problema e dos parâmetros. A meta-heurística está situada em um nível superior e para guiar a busca por soluções melhores, ela seleciona as heurísticas subordinadas, que por sua vez, utilizam técnicas mais específicas para explorar o espaço de soluções ou para melhorar soluções candidatas. A cada iteração, uma ou várias heurísticas subordinadas são aplicadas em sequência ou em paralelo. Cada heurística subordinada desempenha um papel distinto na estratégia de busca da meta-heurística. Após a aplicação destas heurísticas, é verificado se o critério de parada foi atingido. Caso não tenham sido, o ciclo se repete.

Figura 14 – Ciclo meta-heurístico



Em termos simplificados, as meta-heurísticas realizam uma exploração extensa em um espaço de busca desconhecido, portanto são incapazes de identificar de imediato as melhores soluções. Para contornar essa limitação, empregam estratégias específicas tanto na heurística principal quanto nas subordinadas, a fim de mapear áreas promissoras dentro desse espaço e escapar de ótimos locais (processo de diversificação). Assim, a informação acumulada ao longo do processo de exploração, que inclui os lugares visitados (soluções testadas) e a respectiva qualidade (*fitness*) das soluções avaliadas, orienta o algoritmo para tomar decisões direcionadas e mais assertivas.

Embora muitas meta-heurísticas tenham sido inspiradas por padrões observados na natureza ou em comportamentos humanos, como a Otimização por Colônia de Formigas (*Ant Colony Optimization*), Algoritmos Genéticos (*Genetic Algorithms*) e Algoritmos Evolutivos (*Evolutionary Algorithms*), também existem aquelas que não seguem esse paradigma. Entre elas estão a Busca Local Iterada (*Iterated Local Search*), a Busca Dispersa (*Scatter Search*) e a Busca Local Guiada (*Guided Local Search*) (utilizada neste trabalho), que utilizam estratégias mais voltadas à exploração sistemática e combinatória do espaço de soluções, sem a necessidade de uso de alguma inspiração biológica ou social.

2.5.2.4 Meta-heurística *Guided Local Search* (GLS)

Conforme proposto por Voudouris (1997), a GLS, modifica a função objetivo original ao incorporar penalidades em atributos específicos das soluções. Essa estratégia, baseada na aplicação de penalidades, amplia a capacidade e o incentivo para explorar novas áreas do espaço de busca, facilitando a identificação de soluções ótimas ao desencorajar a repetição de escolhas

subótimas anteriormente realizadas.

A GLS inicia seu processo com a geração de uma solução inicial, obtida por meio de uma heurística construtiva. Posteriormente, são estabelecidas as estruturas de dados necessárias para registrar as penalidades e definir os critérios para sua aplicação. Além disso, atributos específicos do problema em questão são identificados e definidos. Por exemplo, no caso do PCV, um atributo relevante é a existência de uma rota potencial entre duas cidades. Nesse contexto, a GLS atribui um custo e uma penalidade a cada atributo identificado. Os custos são geralmente estabelecidos com base nos termos e coeficientes presentes na função objetivo do problema. Neste exemplo do PCV, o custo do atributo pode ser determinado pela distância entre as cidades (VOUDOURIS; TSANG, 1999).

Os valores das penalidades são inicializados em zero e são incrementados somente quando a busca local alcança um ótimo local. Para isso, a GLS emprega uma função de custo que, ao encontrar um ótimo local, é ajustada para cima, aumentando as penalidades sobre as características definidas. Dessa forma, dado uma função objetivo que associa cada solução candidata a um valor numérico real, a GLS define uma função h que é utilizada na busca local, substituindo a função objetivo original g (VOUDOURIS; TSANG, 1999). Essa função de custo é chamada de função de custo aumentada e é definida da seguinte forma:

$$h(s) = g(s) + \lambda \sum_i^i p_i I_i(s), \quad (2.17)$$

onde s é uma solução candidata, λ é um parâmetro utilizado para ajustar a intensidade da busca por soluções. Um valor alto de λ resultará em uma busca mais diversificada, enquanto um valor baixo resultará em uma busca mais focada ou intensa, i é o contador utilizado para percorrer todos os atributos a serem penalizados, p_i é a penalidade associada ao atributo i (dado que todos os p_i são inicializados com 0) e $I_i(s)$ é um valor binário (0 ou 1) que indica se a solução s apresenta o atributo i ou não.

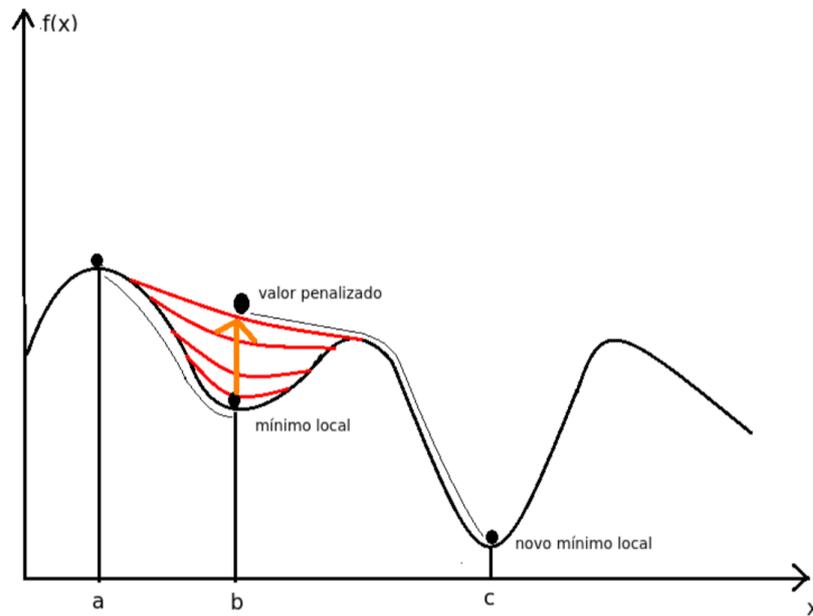
A GLS guia as heurísticas de busca local evitando ótimos locais por meio da penalização na função de custo dos atributos selecionados. Para isso, a seleção de atributos para penalização busca identificar e penalizar de forma prioritária “características desfavoráveis” ou “de grande importância” associadas ao custo global, após se atingir um ótimo local. Características com custo mais elevado impactam diretamente o custo global, sendo importante considerar o valor atual da penalização do atributo. Para tanto, utiliza-se a função $util_i$ que é responsável por penalizar o atributo i individualmente, levando em consideração tanto o custo do atributo quanto a penalidade já aplicada a ele. Essa função não apenas determina se um atributo deve ser penalizado, mas também prioriza a penalização de atributos que têm maior impacto no custo total da solução, especialmente aqueles com custos elevados e com penalidades baixas. Dessa forma, a função $util_i$ guia a escolha de quais características devem ser penalizadas com mais intensidade, ajudando a diversificar a busca e evitando ficar preso em ótimos locais (Figura 15) (VOUDOU-

RIS; TSANG, 2003). Definida a partir de um ótimo local s' , a função $util_i$ assume a seguinte forma:

$$util_i(s') = I_i(s') \frac{c_i}{1 + p_i}, \quad (2.18)$$

onde c_i é o custo associado e p_i é o valor da penalização atual do atributo i . O objetivo desta equação é que se um atributo i não contribuir para a formação do ótimo local, indicado por I_i , então não há benefício em aplicar penalizações sobre ele. A lógica é que, quanto maior o custo c_i de um atributo i , mais benéfica é a sua penalização. Entretanto, à medida que um atributo é penalizado repetidamente, a utilidade de se fazer isso diminui progressivamente. Dentro de uma solução ótima local, os atributos que apresentam os maiores benefícios de penalização, ou seja, aqueles com as maiores utilidades, serão os alvos das penalizações. Sempre que um atributo i sofre uma penalização, o seu valor de penalização p_i é incrementado em uma unidade (VOUDOURIS; TSANG, 2003).

Figura 15 – Deformação do espaço de soluções causada pela GLS



Fonte: Silva e Silva (2015)

O parâmetro λ se caracteriza por ser o único valor que necessita ser calibrado na aplicação da GLS. Ele pode ser calculado levando em conta os próprios resultados do problema, sendo feito após identificar o primeiro mínimo local e antes de começar a penalizar as características ou atributos, o que possibilita um processo dinâmico e adaptativo. Para aplicar essa abordagem, é feito o uso de uma constante α , que não depende da instância do problema. A fórmula empregada para calcular essa relação é dada pela seguinte equação:

$$\lambda = \alpha \frac{g(s')}{\text{quantidade de atributos em } s'}, \quad (2.19)$$

onde g é a função objetivo e s' é um ótimo local encontrado. Ajustes no parâmetro α permitem encontrar um valor de λ adequado para diferentes situações, o que simplifica a calibragem do método (VOUDOURIS; TSANG, 2003). O pseudo-código da meta-heurística *guided local search* pode ser visto no Apêndice F.

Voudouris e Tsang (1999) demonstraram que a GLS é um método eficiente e eficaz para resolver o PCV, fornecendo resultados competitivos e até superiores em comparação com outras técnicas de otimização geral clássicas, como o *Simulated Annealing* e a *Tabu Search*. Especialmente quando combinada com a *Fast Local Search* (FLS) e o método 2-Opt, devido à sua fácil configuração e ajuste de apenas um parâmetro. Os autores mostraram que essa combinação, além de ser superior aos métodos clássicos, também é comparável ou melhor que algoritmos avançados como o *Lin-Kernighan* Iterado e a Busca Local Genética, tornando a escolha da GLS uma opção sólida para resolver o PCV.

2.6 ALGORITMO DE DIJKSTRA

O Algoritmo de Dijkstra, proposto por Edsger Wybe Dijkstra (1959), é uma técnica para resolver o problema de caminhos mínimos (*Shortest Path Problem* (SPP)) de fonte única em grafos ponderados, onde os pesos das arestas representam distâncias ou custos não negativos. Este algoritmo encontra o caminho mais curto entre um vértice de origem e todos os outros vértices em um grafo.

Sua definição é composta por um grafo ponderado $G = (V, E)$, onde V representa um conjunto finito de vértices e E um conjunto de pares ordenados de vértices (arestas). Cada aresta em G é representada por um par ordenado (u, v) , indicando que a aresta inicia no vértice u e termina no vértice v . As arestas do grafo são ponderadas e não negativas, representadas pela função de peso $w(u, v) \geq 0$.

Um caminho em G é definido como uma sequência de vértices $P = \langle v_0, v_1, \dots, v_k \rangle$ de tal forma que cada par consecutivo (v_i, v_{i+1}) forma uma aresta do grafo, para todo i no intervalo de 0 a $k - 1$. Portanto, P é um caminho que vai do vértice v_0 ao vértice v_k . O peso total de um caminho P é a soma dos pesos de todas as arestas que compõem o caminho, expresso por:

$$w(P) = \sum_{i=0}^{k-1} w(v_i, v_{i+1}) \quad (2.20)$$

Ao comparar dois caminhos P e Q que ligam o mesmo ponto de partida v_0 ao ponto de chegada v_k , afirma-se que P é menor que Q se o peso total de P , $w(P)$, for menor que o peso total de Q , $w(Q)$. A distância de um vértice u a um vértice v em G , denotada por $d(u, v)$, é definida como o menor peso dentre todos os caminhos possíveis de u para v . Caso não exista nenhum caminho de u para v , a distância é considerada infinita, simbolizada por $d(u, v) = \infty$. Uma versão simplificada do algoritmo de Dijkstra pode ser vista no Apêndice G.

O uso do algoritmo de Dijkstra neste estudo se revela fundamental, pois possibilita calcular as menores distâncias entre todos os pares de pontos de coleta possíveis de serem percorridas, levando em consideração as restrições físicas e operacionais do armazém. Esse método garante que as rotas otimizadas respeitem a configuração real do espaço e as limitações de movimentação dos robôs, evitando áreas restritas e obstáculos. Sua capacidade de lidar com cenários reais, como os observados neste estudo, o torna adequado para fornecer as informações que alimentam tanto os métodos exatos quanto as meta-heurísticas aplicadas.

3 PROCEDIMENTOS METODOLÓGICOS

O presente estudo explora o desenvolvimento e implementação de uma ferramenta de otimização do processo de *picking* dentro de um contexto de colaboração humano-robô. Para este desenvolvimento, foram seguidas as etapas da metodologia científica de Modelagem e Simulação no contexto de uma pesquisa empírica quantitativa normativa. Segundo Cauchick-Miguel (2019), uma pesquisa empírica quantitativa normativa tem como objetivo desenvolver políticas, estratégias e ações que aprimorem o estado atual com base no uso de modelos, de modo a permitir a identificação de uma solução para o problema de interesse, a qual pode ser gerada a partir do emprego de modelos de otimização ou programação matemática.

A abordagem de Modelagem e Simulação no contexto da Pesquisa Operacional toma por base macro-etapas bem estabelecidas (WINSTON; GOLDBERG, 2004; FILHO, 2008b; HILLIER; LIEBERMAN, 2013), tendo sido utilizadas para definição do método de trabalho adotado. A sequência das etapas do método é apresentada na Figura 16.

Figura 16 – Fluxogramas contendo as macro-etapas do método de trabalho



Fonte: O Autor (2024)

O método de trabalho é caracterizado por quatro macro-etapas: Planejamento, Modelagem, Experimentos de Simulação e Recomendações. Cada uma destas macro-etapas são detalhadas na sequência.

3.1 MACRO-ETAPA DE PLANEJAMENTO

A macro-etapa de planejamento tem como objetivo central reunir e estruturar as informações necessárias para a criação do modelo conceitual, que servirá como a base lógica para o desenvolvimento do modelo computacional e das análises subsequentes. Nessa etapa, busca-se compreender o problema e o contexto operacional, coletar informações gerais e qualitativas necessárias para o planejamento do modelo conceitual.

3.1.1 Definição do problema

Em armazéns tradicionais que não adotam automação, o processo de coleta geralmente é conduzido com o auxílio de um dispositivo coletor de dados ou outra tecnologia mais avançada com o mesmo propósito. As ordens de separação são geradas por um sistema de gerenciamento de armazéns (WMS) de acordo com as estratégias de coleta estabelecidas pela empresa (subse-

ção 2.1.2). Posteriormente, os operadores têm acesso e interagem com essas tarefas por meio desses dispositivos.

Ao iniciar uma ordem de coleta, os operadores percorrem os diversos endereços com um carrinho, paleteira ou recipiente para temporariamente armazenar os produtos coletados. As informações sobre os itens a serem coletados e suas quantidades são exibidas diretamente no dispositivo de coleta. O término do processo de coleta é indicado quando o último item for coletado.

Após concluir a etapa de coleta, os itens geralmente são direcionados para áreas específicas, onde são submetidos a processos como fracionamento, consolidação ou preparação para expedição, dependendo dos procedimentos realizados no armazém. Para facilitar a comunicação, neste trabalho, adotou-se o termo “depósito” para se referir a esses locais. Uma característica desses processos de coleta é a exigência de serem percorridas distâncias significativas entre a área de armazenagem e os depósitos.

Esta movimentação extensa e intensiva dos operadores entre as áreas de coleta e os depósitos, é considerada uma atividade não produtiva. Isso porque, embora os deslocamentos sejam necessários ao processo, eles não agregam valor direto ao produto ou ao serviço oferecido. Além disso, esses deslocamentos demandam esforço físico excessivo e tempo, no qual os operadores poderiam estar dedicados a tarefas produtivas, como a própria coleta dos produtos.

Nos últimos anos, com o avanço das tecnologias de automação, tem-se explorado a colaboração entre robôs moveis e operadores humanos para otimizar o processo de *picking*. Nessa abordagem, os operadores humanos são acompanhados por um AMR ou AGV ao longo de todo o percurso de coleta. Uma vez concluída a separação de todos os itens solicitados, os robôs assumem a responsabilidade de transportar os produtos coletados de volta ao depósito, permitindo que os operadores humanos permaneçam na área de armazenamento/separação. Neste momento, ao iniciar uma nova tarefa, um novo robô é solicitado, que se desloca do depósito para encontrar o operador no primeiro ponto de armazenamento da nova tarefa. Para o operador humano, o deslocamento é do último ponto da tarefa anterior para o primeiro ponto da nova tarefa. Esse modelo colaborativo humano-robô visa a facilitar o transporte dos itens coletados, eliminando a necessidade de retornos frequentes ao depósito e permitindo que os operadores se concentrem primordialmente na coleta de novos pedidos.

Apesar das vantagens potenciais, a implementação desse processo não garante, por si só, ganhos significativos de eficiência. A criação de rotas eficazes para o *picking*, que minimize o tempo de deslocamento dos operadores e maximize a produtividade, exige uma abordagem de otimização. No contexto da colaboração humano-robô, esse desafio de roteamento é potencializado pela necessidade de adaptar as rotas não apenas à disposição física do armazém, mas também à dinâmica das operações conjuntas entre humanos e robôs.

Dessa forma, o problema que se coloca é a necessidade de desenvolver uma ferramenta

computacional que, utilizando técnicas de otimização combinatória, seja capaz de propor rotas eficientes para o processo de *picking* em um ambiente de colaboração entre humanos e robôs. A ferramenta deverá não apenas propor rotas de coletas que minimizem o tempo, mas também se adaptar a diferentes *layouts* de armazéns, gerando resultados em um tempo hábil de modo a permitir sua aplicação em contextos reais.

3.1.2 Coleta de informações

A coleta de informações é realizada com o objetivo de construir o modelo conceitual, que serve de base para a ferramenta de otimização. As informações levantadas refletem aspectos qualitativos do ambiente operacional, necessários para entender a dinâmica entre os operadores humanos, os robôs móveis e o armazém como um todo.

Entre as principais informações a serem coletadas, destaca-se o *layout* do armazém, essencial para modelar os deslocamentos necessários durante a operação de *picking*. São levantadas informações sobre a distribuição dos corredores e zonas de armazenamento e endereços de coleta. Adicionalmente, são identificados obstáculos no armazém que poderiam restringir ou impedir a movimentação eficiente dos robôs.

A análise das tecnologias utilizadas na gestão do armazém também é um ponto central. Avalia-se como os robôs podem se integrar com sistemas como o WMS para executar rotas de coleta e como eles se sincronizam com os operadores humanos. Além disso, verifica-se quais informações relacionadas ao processo de *picking* estão disponíveis, incluindo o formato e a acessibilidade desses dados.

3.1.3 Modelo Conceitual

O modelo conceitual reflete as etapas operacionais da ferramenta de otimização. Ele traduz as operações realizadas em uma sequência lógica e estruturada, desde a entrada de dados até o retorno dos resultados. Essa representação busca traduzir a realidade observada em um formato que seja adequado para realizar a modelagem computacional e permitir a execução dos experimentos, de modo a analisar diferentes estratégias de otimização das rotas de coleta no contexto colaborativo entre humano e robôs.

O modelo conceitual desenvolvido inicia-se com a etapa de importação de dados. Os dados de entrada requeridos incluem endereços de armazenagem, pontos de coleta e obstáculos dentro do armazém, que podem ser importados em diferentes formatos, como por exemplo, *.csv* ou *.xlsx*.

Após a importação dos dados, a ferramenta calcula uma matriz de distâncias entre os pontos de coleta. Essa matriz consiste em armazenar o custo associado aos deslocamentos entre todos os pares de pontos de coleta. Essas distâncias são fundamentais para a otimização subsequente e são armazenadas em um banco de dados, junto com o *layout* do armazém.

Uma vez que o armazém está cadastrado, o sistema recebe uma lista de endereços de armazenagem a serem percorridos de forma ótima, por meio de um *input* manual realizado pelo usuário da ferramenta. Esses endereços são mapeados para pontos de coleta (Apêndice L) e uma matriz de distâncias reduzida, específica destes pontos, é criada, de modo a preparar os dados para o processo de otimização.

A seleção do método de otimização é feita considerando a quantidade de endereços a serem otimizados. Se o número de endereços excede um limite predefinido, a ferramenta opta por uma abordagem heurística, que é mais adequada para grandes volumes de dados e oferece uma solução eficiente em termos de tempo. Caso contrário, um método exato é utilizado, proporcionando uma solução mais precisa.

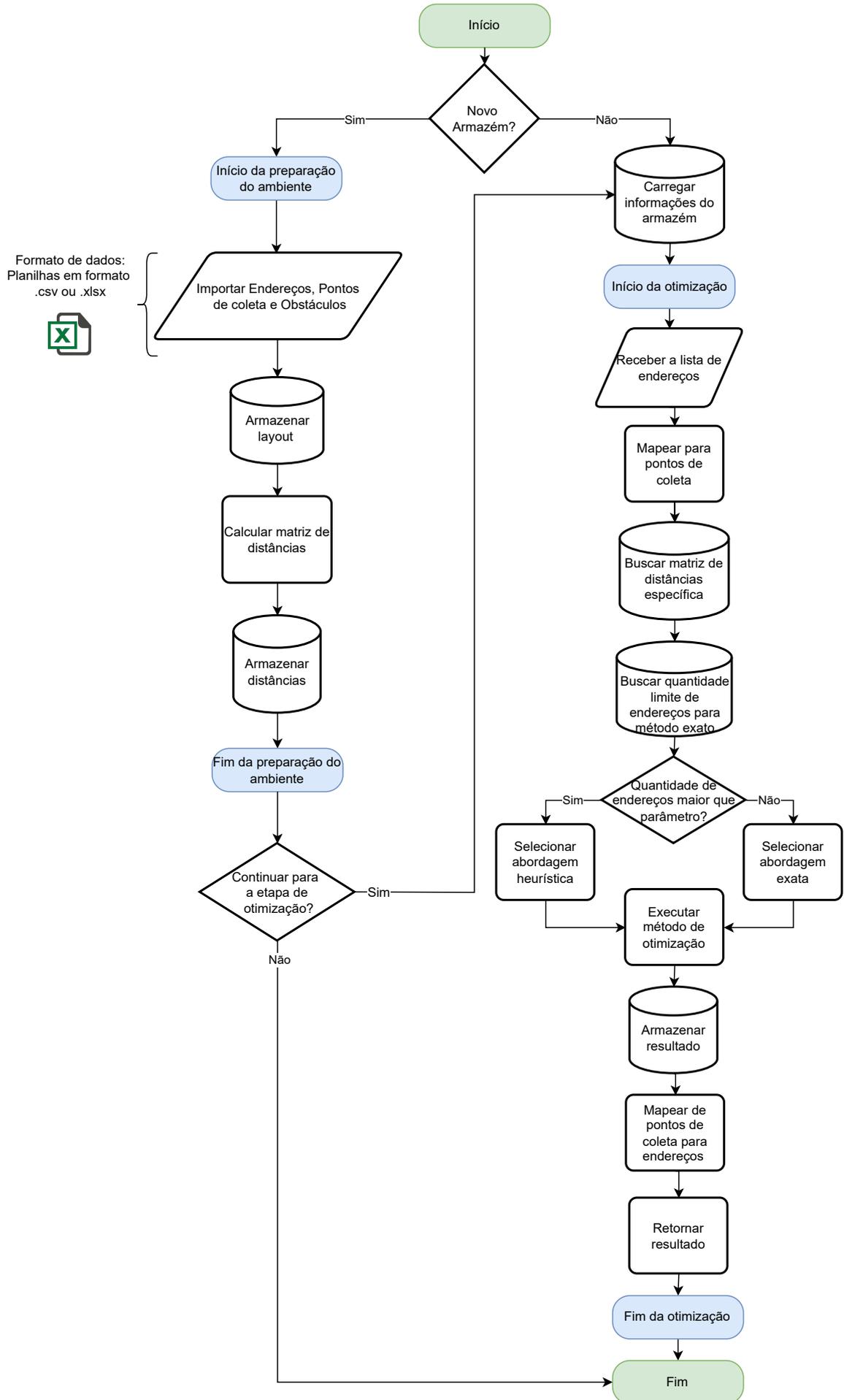
Após a seleção do método, o processo de otimização é executado. Esta fase busca minimizar as distâncias percorridas pelos operadores durante o processo de *picking*. Os resultados são armazenados no sistema e os pontos de coleta são mapeados de volta para os endereços de armazenagem correspondentes.

O processo é finalizado com o retorno dos resultados gerados pela otimização, correspondendo a nova sequência de endereços a ser seguida. Essa informação permite a aplicação prática da rota otimizada por meio da integração com o sistema de gestão do armazém.

Os resultados armazenados, podem ser acessados via uma *interface* visual que permite comparar as rotas recomendadas pelo otimizador com outras rotas fornecidas pelo usuário, geradas a partir do emprego de outras estratégias. Este processo inclui análises e comparações dos resultados alcançados, com o objetivo de verificar se a nova sequência de coleta efetivamente reduz a distância total que os operadores de *picking* precisam percorrer.

A Figura 17 ilustra o modelo conceitual constituído das etapas principais do processo, desde a importação e armazenamento dos dados do armazém até a execução da otimização e o retorno dos resultados. O modelo avalia o desempenho com base em métricas como eficiência computacional e distância total percorrida pelos operadores. Contudo, é importante considerar suas limitações, como a dependência de entradas corretas, desafios em problemas de grande escala e a não atualização de mudanças dinâmicas no *layout* do armazém ou nos obstáculos presentes no ambiente.

Figura 17 – Fluxograma do Modelo Conceitual da Ferramenta de Otimização



3.2 MACRO-ETAPA DE MODELAGEM

O objetivo principal da macro-etapa de modelagem é transformar o modelo conceitual, previamente estabelecido, em um modelo computacional, estruturado para representar virtualmente o armazém e executar as otimizações propostas. Para isso, a modelagem foi estruturada em duas etapas principais: coleta de dados e modelagem computacional.

3.2.1 Coleta de dados

A coleta de dados tem como objetivo fornecer os dados quantitativos e detalhados necessários para a implementação e validação do modelo computacional. Os dados a serem levantados incluem aspectos relacionados ao *layout* físico do armazém e às operações de *picking*. Os dados são fornecidos pelos encarregados da operação do armazém. Planilhas podem ser utilizadas como formato padrão para entrada e armazenamento de dados, permitindo a compatibilidade com a ferramenta desenvolvida. A seguir, são descritas as principais categorias de dados coletados:

3.2.1.1 *Layout* do Armazém e Estrutura de Armazenamento

Uma compreensão do *layout* físico do armazém é fundamental para modelar os deslocamentos necessários durante a operação de *picking*. Neste trabalho, o armazém é representado por um mapa virtual Bidimensional (2D), que reproduz suas características reais, permitindo uma visão estruturada do ambiente. Os dados reunidos possibilitam a coleta de informações sobre:

- a) Distribuição dos corredores e zonas de armazenamento: Identificação dos corredores transversais e longitudinais, prateleiras, e áreas específicas de armazenamento;
- b) Endereços de armazenamento: Mapeamento dos locais de armazenamento de produtos que serve como referência para os pontos de coleta, os quais são utilizados pelo robô para realizar a rota separação;
- c) Localização do depósito central: Definição do ponto central para onde os robôs transportam os itens coletados, representando o ponto final de cada rota de coleta conforme a estratégia utilizada;
- d) Localização do ponto de partida: Definição do ponto de partida onde os robôs iniciam o deslocamento, representando o ponto inicial de cada rota de coleta;
- e) Identificação de obstáculos: Mapeamento dos obstáculos no armazém que impedem ou restringem a movimentação dos robôs;
- f) Distâncias entre estruturas: Medições da largura dos corredores, comprimento das prateleiras e porta-paletes e distâncias entre pontos de referência;

- g) Coordenadas 2D e disposição física das estruturas mapeadas: Localização espacial de cada estrutura em um plano bidimensional, considerando sua posição física e posicionamento dentro do armazém.

3.2.1.2 Estratégias Atuais de Coleta

O entendimento das estratégias de coleta atualmente adotadas no armazém servem como referência para avaliar os resultados da metodologia proposta. A análise dessas estratégias oferece uma base para a comparação dos benefícios das rotas otimizadas, permitindo identificar os ganhos potenciais ao substituir as rotas utilizadas pelas rotas otimizadas. Para isso, foram consideradas as seguintes informações:

- a) Estratégias de Roteamento: Descrição das políticas de roteamento adotadas, indicando a lógica na qual os operadores se movem pelos corredores/endereços e coletam os itens;
- b) Métricas de Desempenho: Dados históricos de desempenho, como distância média percorrida por coleta, tempo total para completar uma ordem de *picking*;
- c) Histórico de execução de *pickings* anteriores: Dados históricos que possibilite reconstruir a ordem de visita dos endereços em coletas realizadas anteriormente, categorizado por operador. Esses dados servem como base comparativa para avaliar as diferenças em relação às rotas otimizadas.

3.2.2 Modelagem Computacional

O desenvolvimento do modelo computacional apresentado neste estudo baseia-se no modelo conceitual descrito na Seção 3.1.3 e se divide em duas etapas principais. A primeira etapa envolve a preparação do ambiente, que inclui a construção de uma matriz de distâncias entre os pontos de coleta. A segunda etapa foca na otimização, que envolve a montagem de uma matriz de distâncias específica derivada da matriz principal, seguida pela escolha e aplicação dos métodos de otimização. Detalhes dessas etapas são apresentados a seguir.

3.2.2.1 Construção da Matriz de Distâncias

A construção da matriz de distâncias começa com o uso das coordenadas cartesianas e a identificação dos objetos associados a elas, extraídos do modelo virtual 2D que representa a configuração espacial real do armazém. Nesse modelo, cada célula equivale a uma unidade de espaço no armazém e pode representar áreas livres, nas quais o movimento é permitido, ou obstáculos e estruturas de armazenagem, onde o movimento é proibido, isso faz com que as rotas respeitem as restrições reais do ambiente.

Para calcular as distâncias entre cada par de pontos de coleta, emprega-se o algoritmo de Dijkstra (Seção 2.6), aplicado à matriz bidimensional do armazém. Esse método permite que

as áreas restritas sejam “desviadas” pelo algoritmo durante o cálculo da rota, garantindo que nenhum caminho proposto passe por tais células. Em vez de utilizar arestas de um grafo, o cálculo é realizado com base nas adjacências diretas entre células, considerando apenas movimentos horizontais e verticais. Essa restrição de movimento reflete as capacidades de navegação dos robôs no armazém, garantindo que as soluções de roteamento propostas sejam viáveis dentro das limitações operacionais existentes. Detalhes de como a matriz de distâncias é construída podem ser vistos no Apêndice J.

3.2.2.2 Método de Otimização

Com os dados do armazém carregados, o método de otimização iniciará recebendo uma lista de endereços de coleta que precisam ser visitados durante o processo de *picking*. Primeiramente, com base no cadastro, esses endereços devem ser mapeados para seus respectivos pontos de coleta (Apêndice L). Somente os pontos de coleta identificados nesta fase devem ser usados para compor a matriz de distâncias específica da instância atual a ser otimizada.

Uma vez que essa matriz de distâncias esteja montada, ela é disponibilizada para a etapa de otimização, que emprega dois métodos distintos: um exato (Seção 2.5.1) e um heurístico (Seção 2.5.2). A seleção entre os métodos de otimização será determinada automaticamente com base no tamanho da instância do problema, definido pela quantidade de endereços a serem considerados no processo de otimização. Esse valor será definido por um parâmetro configurável que estabelecerá um limiar numérico delimitando o uso de cada método, de modo que, se o número de endereços é inferior ou igual ao limiar, a ferramenta optará pelo método exato, caso contrário, o método heurístico será empregado. Esta abordagem permite que a ferramenta se ajuste automaticamente à complexidade da tarefa de otimização, permitindo balancear a precisão e a eficiência de tempo, quando necessário, características essenciais para aplicações práticas.

A abordagem de otimização a ser adotada, seja o PCV ou o PCVA, é determinada com base em parâmetros fornecidos, isto é, a estratégia de otimização a ser utilizada e o controle do fluxo das tarefas de coleta anteriores. No método exato, os problemas são resolvidos utilizando a técnica *Branch and Cut*, que é aplicada na solução do modelo de Programação Linear Inteira Mista (PLIM), conforme detalhado na Seção 2.5.1.2. Quanto ao método heurístico, emprega-se a meta-heurística GLS, cuja fundamentação teórica sobre o funcionamento e sua escolha é apresentada na Seção 2.5.2.4.

3.2.2.3 Visualização dos resultados

Para facilitar a apresentação e análise comparativa dos resultados, desenvolveu-se uma interface web que inclui uma visualização 2D do *layout* do armazém e um módulo para análise quantitativa dos resultados. Essa interface tem como objetivo facilitar a comparação entre as rotas sugeridas pelo sistema proposto e aquelas atualmente utilizadas pelos armazéns. O foco da análise é dado à avaliação das melhorias alcançadas, quantificando a redução na distância

de coleta e o potencial aumento na eficiência. Mais detalhes sobre a visualização virtual do armazém podem ser encontrados no Apêndice I.

3.2.2.4 Validação do Modelo

A validação do modelo é realizada para verificar se os métodos implementados funcionam conforme o esperado e para avaliar a sensibilidade do parâmetro de tempo de execução da abordagem heurística. Para isso, são utilizadas instâncias disponibilizadas em bibliotecas e bases de dados para o PCV, permitindo testar e comparar os resultados obtidos com valores previamente validados e já conhecidos na literatura.

3.3 MACRO-ETAPA DE EXPERIMENTOS DE SIMULAÇÃO

Nesta seção, são descritos os métodos empregados para avaliar as estratégias de otimização propostas neste trabalho. Os experimentos foram organizados em três categorias: i) experimentos de validação, conforme descrito anteriormente, que verificam a precisão e a robustez dos métodos implementados, utilizando instâncias conhecidas da literatura; ii) experimentos de avaliação de desempenho, que utilizam dados coletados de um ambiente real de armazenagem para comparar a eficácia e eficiência das abordagens exata e heurística; iii) experimentos aplicados, que abordam e solucionam problemas práticos relacionados às rotas de coleta em um processo de colaboração humano-robô, baseados nos dados disponibilizados por uma empresa real.

3.3.1 Definição de Cenários de Avaliação

Os cenários de avaliação fazem parte dos experimentos aplicados e foram criados utilizando instâncias obtidas de uma operação de *picking* real, que contém a sequência de coleta executada pelos operadores. Cada cenário foi estruturado para avaliar as estratégias de otimização propostas, sendo divididos em dois tipos principais:

- a) Cenário de Coletas Independentes: Neste cenário, tanto o robô quanto o operador humano iniciam e encerram suas atividades na área de depósito. Cada rota de coleta é planejada de forma independente, refletindo um processo operacional comumente utilizado pelas empresas;
- b) Cenário de Coletas Encadeadas: Neste cenário, o robô retorna ao depósito enquanto o operador humano permanece no local do último ponto de coleta, iniciando a próxima rota a partir desse ponto. Essa abordagem reduz o deslocamento do operador, sendo mais alinhada às estratégias de colaboração humano-robô.

3.3.2 Critérios de Desempenho

Para avaliar o funcionamento e a capacidade de uso real da ferramenta empregando os métodos utilizados, são explorados dados coletados a partir da execução das abordagens exata

e heurística. As abordagens são comparadas em termos de eficácia e eficiência operacional, por meio da medição do tempo de execução do algoritmo, do tempo necessário para a criação da matriz de distâncias, e do tempo total envolvido, que engloba tanto a execução do algoritmo quanto a preparação prévia dos dados. Adicionalmente, é avaliada a qualidade das rotas otimizadas com base na distância percorrida pelo robô, que representa o percurso total necessário para completar uma tarefa, considerando os deslocamentos entre o depósito e os pontos de coleta, e mede a eficiência das rotas ao minimizar o transporte dos itens até o depósito; além disso, considera-se também a distância percorrida pelo operador humano, que toma como base os deslocamentos realizados para concluir uma tarefa, visando reduzir o esforço físico do operador, o tempo gasto e a distância percorrida nas operações.

3.3.3 Análise dos Resultados

A análise dos resultados obtidos nos experimentos de simulação é conduzida para avaliar a eficiência do modelo computacional desenvolvido em relação à política de roteamento implementada pelo atual sistema WMS da empresa em avaliação. Os resultados são organizados e comparados com base nas métricas de desempenho definidas, como distância total percorrida, tempo de execução individual das abordagens de otimização avaliadas e tempo total de execução.

3.4 MACRO-ETAPA DE RECOMENDAÇÕES

A partir dos resultados dos experimentos aplicados, é conduzida uma análise com o objetivo de comparar as estratégias adotadas. A análise avalia as diferentes abordagens de otimização utilizadas, identificando aquelas que apresentaram maiores reduções nos deslocamentos. Assim, os resultados obtidos podem servir de base para a recomendação da estratégia de otimização de rotas com melhor desempenho observado, podendo contribuir para a melhoria dos processos logísticos envolvidos.

4 EXPERIMENTOS E RESULTADOS

O objetivo deste trabalho foi desenvolver uma ferramenta de otimização para aprimorar o processo de *picking* realizado em armazéns, com suporte de AGVs ou AMRs. O cenário considerado envolve um conjunto de ordens de coleta, assumindo que cada SKU está localizado em uma posição específica de armazenamento. O processo de coleta de um pedido consiste em visitar e coletar itens a partir de um conjunto de posições de *picking*. Um *tour* viável é definido como aquele que visita as posições de *picking*, garantindo que cada robô seja carregado com apenas uma tarefa de coleta. Dentro do conjunto de *tours* viáveis, busca-se aquele que minimiza a distância total percorrida pelo operador.

Para a realização de todos os experimentos, foi utilizado um computador com sistema operacional Linux (Arch Linux), equipado com 32 GB de memória RAM e com um processador Intel(R) Core(TM) i7-277700HQ de 8 núcleos (sendo 4 físicos), com uma velocidade de *clock* de 2.80 GHz e um cache de 6144 KB.

Quanto à implementação computacional da ferramenta, utilizaram-se as linguagens de programação *Kotlin* para o *back-end* e *JavaScript* para o *front-end*. O armazenamento das informações é feito por meio do sistema de gestão de banco de dados Postgres. Para executar as abordagens de otimização avaliadas neste estudo, utilizou-se a *OR-Tools*, uma biblioteca desenvolvida pela Google para auxiliar na solução de problemas de otimização combinatória (PERRON; FURNON, 2024). Para a aplicação do método exato foi utilizado o módulo “MPSolver”, enquanto o módulo “CP-SAT” foi empregado para o método heurístico.

Os experimentos foram realizados em três etapas distintas. A primeira etapa de experimentos tem como objetivo validar a ferramenta de otimização desenvolvida, por meio do uso de instâncias da literatura disponibilizadas pelas bibliotecas *TSPLIB* (HEIDELBERG, 2023) e da *TSP Data* da *University of Waterloo* (WATERLOO, 2023). A qualidade de cada solução final é avaliada e comparada com valores disponibilizados nos repositórios das respectivas bibliotecas, verificando se as abordagens desenvolvidas funcionam e geram resultados de acordo com o esperado. Além disso, o tempo necessário para gerar a resposta foi utilizado como base para ajustar os parâmetros de tempo, que determinam o critério de parada da meta-heurística GLS.

A segunda etapa de experimentos consiste na avaliação de desempenho da ferramenta de otimização quando aplicada em um ambiente real de armazenagem. Nessa etapa, são gerados cenários aleatórios para o caso real de um armazém, a serem utilizados para avaliar as abordagens de otimização que compõem a ferramenta desenvolvida, isto é, as abordagens exata e a heurística. Aspectos como a qualidade da solução final e o tempo necessário para gerar a resposta da solução são comparados e avaliados, dado que esses fatores são importantes do ponto de vista da aplicabilidade prática da ferramenta de otimização.

A terceira etapa consiste na realização de experimentos práticos voltados à otimização

das rotas. Estes experimentos utilizam cenários reais de operação, contendo as sequências de coleta realizadas pelos operadores, com o objetivo de comparar a otimização individual das rotas dos robôs e dos colaboradores humanos. Durante essa fase, diferentes estratégias para execução das sequências de coleta são propostas e avaliadas. Por meio da realização destes experimentos, busca-se identificar, dentre as estratégias sugeridas, aquela que apresenta o melhor desempenho segundo os critérios analisados.

4.1 EXPERIMENTOS DE VALIDAÇÃO

Com o objetivo de verificar se as implementações dos métodos de otimização empregados na ferramenta computacional desenvolvida funcionam conforme o esperado, e também avaliar a sensibilidade do parâmetro de tempo de execução do método heurístico, foram utilizadas instâncias assimétricas 2D das bibliotecas *TSPLIB* e *TSP Data*. O uso das instâncias disponibilizadas por estas bibliotecas possibilitou testar e comparar os resultados encontrados pela ferramenta computacional desenvolvida com valores validados e já conhecidos na literatura. As instâncias utilizadas incluem, *Wi29*, *DJ38*, *Berlin52*, *Pr76* e *Pr136*, extraídas da *TSPLIB* (HEIDELBERG, 2023) e da *TSP Data* da *University of Waterloo* (WATERLOO, 2023). É importante destacar que não foram avaliadas instâncias maiores por elas não representarem a realidade comumente observada em operações de separação realizadas em armazéns.

Para quantificar a precisão dos resultados encontrados pelas abordagens de otimização da ferramenta desenvolvida em comparação aos valores ótimos conhecidos, foi utilizada a medida do erro percentual, calculada pela seguinte expressão:

$$\text{Erro} = \left| \frac{\text{valorEncontrado} - \text{melhorValor}}{\text{melhorValor}} \right| \times 100 \quad (4.1)$$

4.1.1 Validação do Método Exato

A Tabela 1 apresenta os resultados obtidos ao resolver as instâncias de validação utilizando a abordagem exata implementada na ferramenta desenvolvida. Todas as instâncias convergiram para o valor ótimo após determinado tempo. Esses resultados demonstraram que a ferramenta desenvolvida estava fornecendo valores confiáveis, uma vez que, após processar os dados de entrada, otimizar e processar os dados de saída, os resultados obtidos coincidiam com aqueles previamente validados pela literatura.

Tabela 1 – Resultados das Instâncias de validação executando a abordagem exata

Instância	Tamanho	Valor Ótimo	Valor encontrado	Tempo (s)	Erro (%)
<i>Wi29</i>	29	27603	27603	227	0,0
<i>Dj38</i>	38	6656	6656	352	0,0
<i>Berlin52</i>	52	7542	7542	7	0,0
<i>Pr76</i>	76	108259	108259	430	0,0
<i>Pr136</i>	136	96772	96772	872	0,0

Em termos de performance, a análise do tempo computacional necessário para encontrar a solução exata revelou que o modelo utilizado não apresentava desempenho adequado às aplicações práticas avaliadas, sobretudo ao lidar com instâncias maiores, como as que podem ocorrer em um contexto real de *picking*.

4.1.2 Validação do Método Heurístico

Para validar o método heurístico implementado na ferramenta desenvolvida e analisar o impacto do parâmetro de tempo que define o critério de parada para a meta-heurística, todas as instâncias foram submetidas a uma série de testes com tempos de execução variados. Os testes começaram com um tempo de execução de 50 milissegundos, sendo incrementados em intervalos de 50 milissegundos até alcançar 1 segundo. Após o primeiro segundo de execução, os incrementos passaram a intervalos de 1 segundo, continuando assim até atingir 20 segundos. A partir de 20 segundos, os incrementos foram ajustados para 25 segundos. A Tabela 2 apresenta uma amostra dos resultados obtidos, mostrando os valores da execução dos três primeiros limites de tempo, para cada instância, após uma única rodada de experimentos.

Tabela 2 – Resultados das Três Primeiras Execuções por Instância na Validação da Heurística

Instância	Tamanho	Valor Ótimo	Valor encontrado	Tempo (s)	Erro (%)
<i>Wi29</i>	29	27603	27750	0,05	0,53
<i>Wi29</i>	29	27603	27750	0,1	0,53
<i>Wi29</i>	29	27603	27750	0,15	0,53
<i>Dj38</i>	38	6656	6808	0,05	2,28
<i>Dj38</i>	38	6656	6808	0,1	2,28
<i>Dj38</i>	38	6656	6660	0,15	0,06
<i>Berlin52</i>	52	7542	8262	0,05	9,55
<i>Berlin52</i>	52	7542	8262	0,1	9,55
<i>Berlin52</i>	52	7542	8262	0,15	9,55
<i>Pr76</i>	76	108259	117588	0,05	8,62
<i>Pr76</i>	76	108259	112140	0,1	3,58
<i>Pr76</i>	76	108259	112140	0,15	3,58
<i>Pr136</i>	136	96772	105768	0,05	9,30
<i>Pr136</i>	136	96772	104992	0,1	8,49
<i>Pr136</i>	136	96772	104992	0,15	8,49

Durante cada execução, o tempo limite foi definido e o método era executado, registrando o resultado obtido para cada tempo limite avaliado. O processo continuava até que o melhor valor conhecido fosse atingido. Este procedimento foi repetido 15 vezes para cada instância. Posteriormente, para cada limite de tempo estipulado, a média dos valores encontrados nas 15 execuções foi calculada, de modo a ajustar a aleatoriedade do método heurístico.

A Tabela 3 resume os resultados de tempo necessários para ser encontrada a solução ótima das instâncias de validação utilizando a abordagem heurística implementada na ferramenta desenvolvida. Todas as instâncias convergiram para o valor ótimo após determinado tempo. Es-

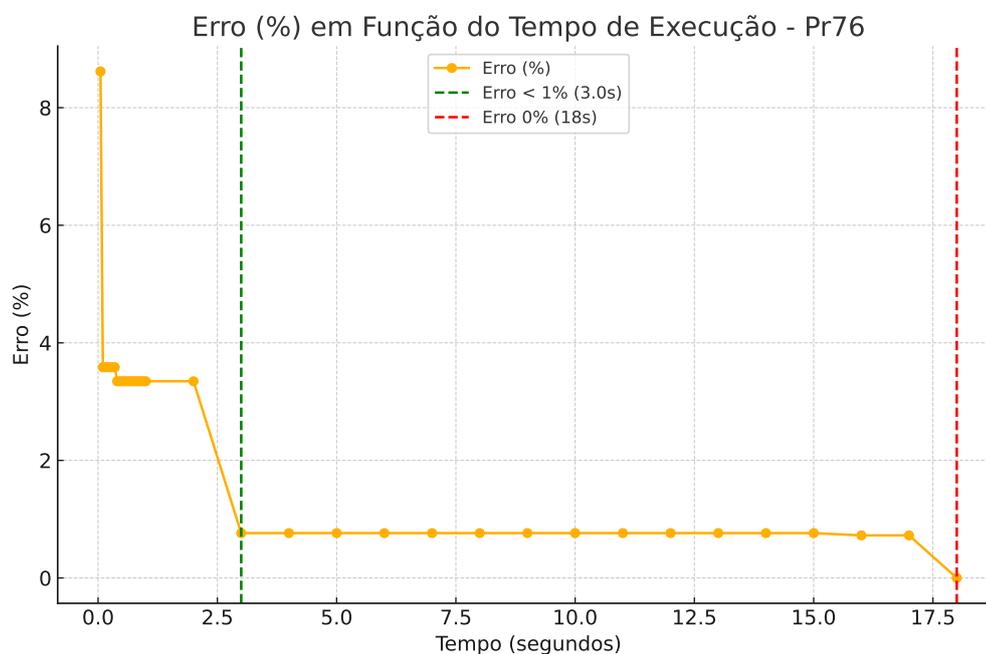
ses resultados demonstram que a ferramenta desenvolvida está fornecendo valores consistentes, uma vez que, após processar os dados de entrada, otimizar e processar os dados de saída, os resultados obtidos coincidem com aqueles previamente validados pela literatura.

Tabela 3 – Resumo dos resultados obtidos das instâncias de validação executando a abordagem heurística

Instância	Tamanho	Valor Ótimo	Valor encontrado	Tempo (s)	Erro (%)
<i>Wi29</i>	29	27603	27603	0,2	0,0
<i>Dj38</i>	38	6656	6656	0,2	0,0
<i>Berlin52</i>	52	7542	7542	0,6	0,0
<i>Pr76</i>	76	108259	108259	18	0,0
<i>Pr136</i>	136	96772	96772	350	0,0

Além de demonstrar que a ferramenta está produzindo valores confiáveis, esse experimento possibilitou a compreensão para a escolha do parâmetro de tempo adequado que define o critério de parada para a execução do método heurístico. Ao analisar o erro em função do tempo de execução, foi possível observar que a maior parte da melhoria na solução ocorre nos primeiros segundos de execução. Para ilustrar essa relação, as Figuras 18 e 19 apresentam gráficos que mostram essa tendência.

Figura 18 – Erro (%) em Função do Tempo de Execução - *Pr76*

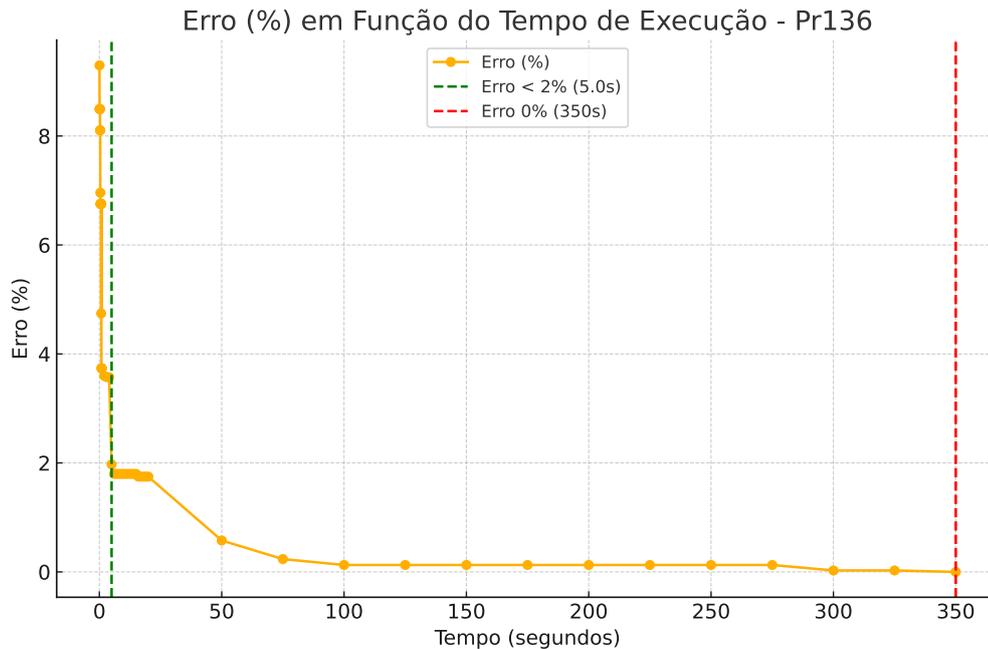


Fonte: O Autor (2024)

Ao analisar o gráfico da Figura 18, observa-se que o erro atinge 0% (linha tracejada em vermelho) em 18 segundos. No entanto, em cerca de 2,6 segundos, o erro percentual já está abaixo de 1% (linha tracejada em verde), mostrando uma capacidade de otimização considerável em relação ao tempo analisado. Para reduzir o erro de menos de 1% para 0%, foi necessário um

tempo adicional de aproximadamente 15,4 segundos (quase 6 vezes mais), o que representa uma melhora na qualidade da solução de apenas cerca de 1% em relação à qualidade da solução de referência. Em outras palavras, em 2,6 segundos, o método atingiu aproximadamente 99% da otimização máxima estimada.

Figura 19 – Erro (%) em Função do Tempo de Execução - *Pr136*



Fonte: O Autor (2024)

Analisando o comportamento do gráfico da Figura 19, observa-se que em cerca de 5 segundos o erro percentual da otimização da instância *Pr136* já está abaixo de 2% (linha tracejada em verde), mostrando uma redução expressiva em um curto período de tempo. Para reduzir o erro de menos de 2% para 0% (intervalo de tempo entre as linhas tracejadas verde e vermelha), foi necessário um tempo adicional de aproximadamente 345 segundos (69 vezes mais), o que representa uma melhoria na qualidade da solução de menos de 2%, em relação ao tempo total.

Essa análise mostra que a maior parte da melhoria na solução ocorre nos primeiros segundos de execução, e esperar mais tempo não gera melhorias proporcionais ao investimento em tempo adicional. Por exemplo, para a instância *Pr136*, onde observa-se que o erro caiu abaixo de 2% nos primeiros segundos de execução do otimizador, o ganho adicional de se dar continuidade ao processo de otimização até atingir 0% é baixo quando comparado ao tempo adicional necessário. Isso reforça a ideia de que esperar um tempo maior pode não justificar os pequenos ganhos na precisão da solução.

Portanto, essa análise sugere que é possível utilizar um limite de tempo dinâmico de acordo com o tamanho da instância. Para instâncias menores, com até 50 pontos, pode-se utilizar uma variação de tempo iniciando em 200 ou 300 milissegundos, chegando a um máximo

de 1 ou 2 segundos. Para instâncias maiores, com até 140 pontos, um tempo de 5 segundos mostrou-se adequado para equilibrar qualidade da solução e tempo de processamento, levando em consideração o uso da ferramenta de otimização em um ambiente real.

4.1.3 Experimentos de Desempenho em Ambiente Real

Neste conjunto de experimentos será tomado como base um ambiente real de armazenagem. O ambiente real avaliado corresponde ao armazém de uma empresa localizada na cidade de Caxias do Sul e que atua como fornecedora de fixadores para fabricantes de diversos segmentos. A sua sede atual ocupa uma área de mais de 4000 m² e possui uma infraestrutura de armazenagem composta por porta-paletes, distribuídos entre 10 corredores e contando com mais de 5000 posições de armazenamento, mapeados em 117 pontos de coleta. Detalhes da representação deste espaço físico podem ser encontrados no Apêndice K.

Para avaliar o funcionamento e a capacidade de uso real da ferramenta de otimização desenvolvida, foram explorados dados coletados a partir da execução da abordagem exata e heurística quando aplicadas na solução de cenários hipotéticos criados com base neste ambiente real de armazenagem.

As abordagens de otimização foram comparadas em termos de eficácia e eficiência operacional, por meio da medição do tempo de execução do algoritmo, do tempo necessário para a criação da matriz de distâncias, e do tempo total envolvido, que engloba tanto a execução do algoritmo quanto a preparação prévia dos dados. Adicionalmente, os dados incluem o número de endereços submetidos à otimização e o valor final da Função Objetivo (FO), que reflete a qualidade da solução encontrada por cada método. Todas as medições de tempo são expressas em milissegundos.

Observa-se que cenários aleatórios foram gerados, sendo cada um analisado separadamente, garantindo a comparabilidade dos resultados. A cada incremento no número de endereços, um novo conjunto de endereços era gerado e aplicado individualmente a cada uma das abordagens. Os experimentos foram realizados assumindo instâncias de diferentes tamanhos, partindo de instâncias com 10 endereços de coleta e, para cada quantidade de endereços, sendo gerados aleatoriamente vinte e um cenários diferentes e submetidos à otimização. Este processo foi repetido sucessivamente, com a quantidade de endereços aumentando de um em um, progressivamente, até alcançar 32 endereços por instância.

Os critérios de parada para os métodos foram estabelecidos com base nos experimentos realizados durante a etapa de validação do modelo (Seção 4.1). Para o método heurístico, o critério de parada foi ajustado de forma dinâmica pela ferramenta, dependendo do tamanho da instância: para instâncias com menos de 20 pontos de coleta, o tempo limite foi fixado em 300 milissegundos; para instâncias entre 20 e 30 pontos de coleta, 1 segundo; entre 30 e 40 pontos de coleta, 1,5 segundos; e, para instâncias maiores que 40 pontos de coleta, o tempo variava entre 2 e 5 segundos, conforme a complexidade do problema. Esta definição contempla cenários

de qualquer tamanho. No caso do método exato, como sua execução se encerra ao encontrar a solução ótima, foi definido um tempo limite de 120 segundos. Caso a solução exata não fosse encontrada dentro desse período, o método era finalizado e retornava o melhor valor encontrado até então.

A análise dos resultados foi realizada com base em valores médios. Inicialmente, foram calculadas médias dos resultados obtidos para cada tamanho de instância, quando empregadas as abordagens de otimização propostas. Em seguida, uma média geral foi obtida a partir das médias desses grupos.

Tabela 4 – Valores Médios das Métricas de Desempenho para as Abordagens de Otimização

Abordagem	Valor FO	Execução (s)	Tempo Criação Matriz (s)	Tempo Total (s)
Exata*	234,60	25,18	1,07	26,25
Heurística	234,31	0,95	1,07	2,02

* Valores médios afetados pelo critério de parada (limite de 120 segundos)

A Tabela 4 reúne os resultados obtidos nos experimentos realizados. Os valores médios da FO são próximos entre as duas abordagens, com a abordagem exata de otimização tendo fornecido o valor médio de 234,60, enquanto a abordagem heurística forneceu o valor médio de 234,31. Isso indica que as abordagens apresentam resultados semelhantes em termos de qualidade da solução.

No entanto, observa-se que o valor da abordagem exata está destacado por um asterisco (*). Esse destaque indica que, em algumas instâncias, o método exato atingiu o critério de parada por tempo máximo (120 segundos) antes de alcançar a solução ótima (ver Tabela 5). Essa interrupção justifica o comportamento inesperado observado, em que o método exato apresentou resultados inferiores aos da abordagem heurística.

Em relação ao tempo de execução, as abordagens apresentam diferença evidente. A abordagem exata levou, em média, 25,18 segundos, enquanto a abordagem heurística executou em média em 0,95 segundos. Essa diferença sugere que, dependendo da urgência operacional, especialmente em situações práticas, e do número de endereços envolvidos, a utilização do método heurístico pode ser mais adequada.

O tempo médio para criação da matriz de distâncias foi igual para ambas as abordagens, com valor de 1,07 segundos. Esse resultado era esperado, já que a construção da matriz é realizada antes dos processos de otimização e independe da abordagem escolhida.

A abordagem heurística, mesmo restrita a um tempo de execução dinâmico de 300 milissegundos a 1,5 segundos (considerando o cenário de maior tamanho), alcança resultados próximos aos da abordagem exata, cujo tempo de execução leva, aproximadamente, em média 25 segundos para completar. Esses resultados sugerem que o método heurístico é eficiente na obtenção de soluções próximas ao ótimo, além de evidenciar que a estratégia de escolha do parâmetro

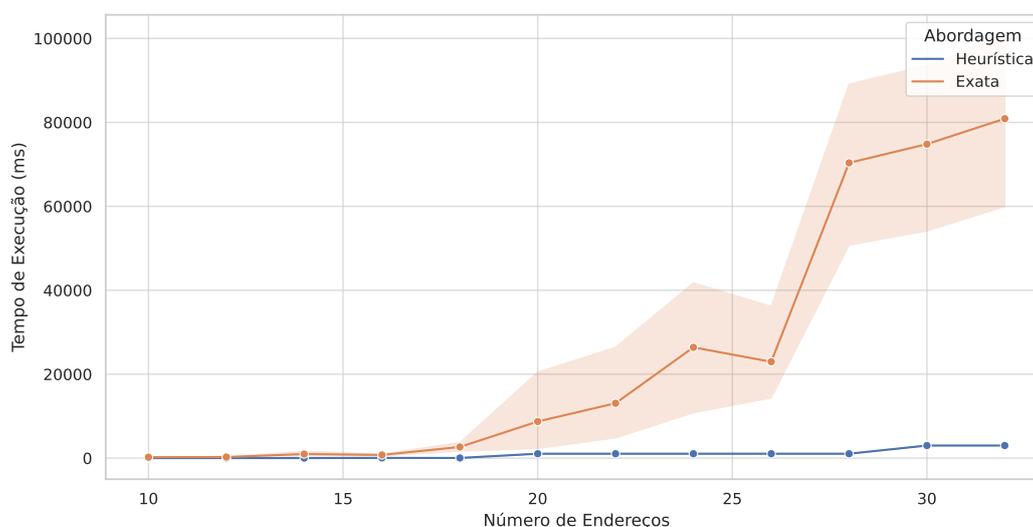
adotado para seleção dinâmica de tempo é adequada, de modo a permitir um balanceamento entre a qualidade de solução e custo computacional.

Tabela 5 – Proporção de interrupções do Método Exato de acordo com o tamanho da instância

Número de Endereços	Interrupções (%)
10	0,00
12	0,00
14	0,00
16	0,00
18	0,00
20	4,76
22	4,76
24	9,52
26	4,76
28	33,33
30	38,10
32	57,14

A Tabela 5 mostra a porcentagem das interrupções do método exato com base na quantidade de endereços existente nas instâncias avaliadas, devido ao processo de otimização exceder o limite de tempo de execução de 120 segundos. Observa-se que há uma tendência de aumento na quantidade de interrupções conforme o número de endereços cresce. Isso sugere que o método exato pode enfrentar dificuldades relacionadas à escalabilidade, mostrando-se não ser adequado para instâncias de tamanhos maiores ou igual a 20, considerando o seu uso em um ambiente real.

Figura 20 – Tempo de Execução (ms) por Número de Endereços

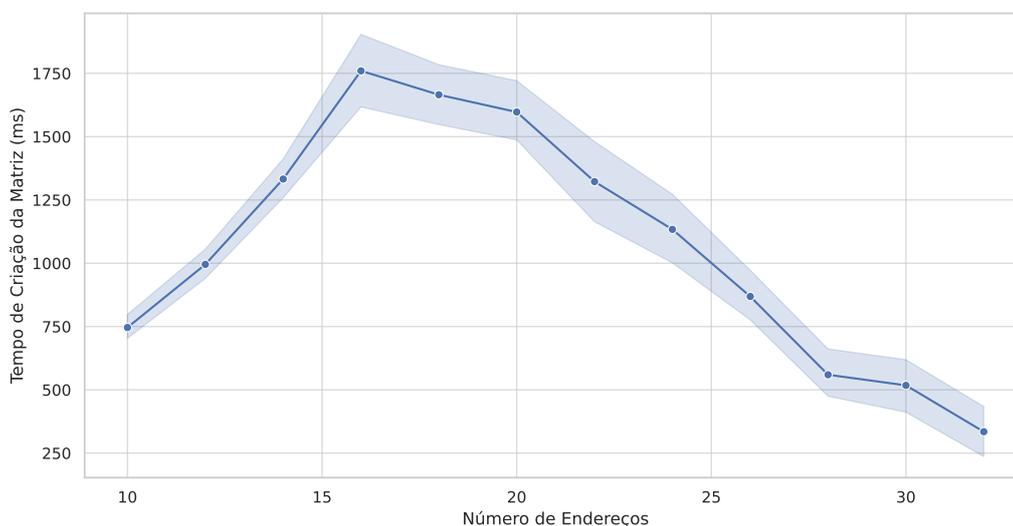


Fonte: O Autor (2024)

No gráfico da Figura 20, observa-se que, para ambas as abordagens, o tempo de execução aumenta à medida que o número de endereços cresce. No entanto, a taxa de crescimento para a

abordagem exata é notavelmente maior em comparação com a abordagem heurística, indicando que o tempo necessário para a execução da abordagem exata aumenta, de maneira acentuada, a partir de instâncias com 18 endereços.

Figura 21 – Tempo de Criação da Matriz (ms) por Número de Endereços



Fonte: O Autor (2024)

Ao analisar o tempo de criação da matriz de distâncias em função da quantidade de endereços presentes nas instâncias avaliadas, inicialmente, espera-se que o tempo aumente proporcionalmente ao número de endereços processados, refletindo o crescimento da complexidade computacional. Entretanto, observa-se no gráfico da Figura 21 que, após um aumento inicial, o tempo de processamento começa a diminuir, sugerindo a influência de mecanismos de cache e otimizações de consulta, tanto no nível do banco de dados quanto do *framework* utilizado para o mapeamento objeto-relacional. Dados frequentemente acessados são armazenados temporariamente em cache, permitindo recuperações mais rápidas em acessos subsequentes, o que pode explicar a redução no tempo observado a partir de um determinado número de endereços que compõem a instância. Além disso, a adaptação dos planos de execução das consultas pelo sistema, com base nas operações anteriores, pode também contribuir para essa melhora no desempenho observado.

No geral, a análise indica que, para instâncias com pequenas quantidades de endereços, a abordagem exata é viável, oferecendo soluções precisas sem comprometer o tempo de execução. Em contraste, para instâncias maiores, a abordagem heurística se mostra eficaz e rápida, com um *trade-off* aceitável com relação à precisão, já que os resultados mostram que ambos os métodos alcançam valores da função objetivo similares.

4.2 EXPERIMENTOS APLICADOS

Para avaliar as estratégias de otimização desenvolvidas, dois cenários distintos foram estabelecidos como referências para a comparação de resultados. Em cada cenário, confronta-se o desempenho da política de roteamento implementada pelo atual sistema WMS da empresa com os resultados derivados do emprego de cada uma das estratégias de otimização implementadas. Entende-se como resultado satisfatório toda e qualquer ação que o sistema de otimização contribuir para a redução da distância percorrida pelos operadores durante o processo de separação de pedidos conduzido pelas empresas.

Para a preparação dos problemas-teste submetidos à otimização e análise subsequente, extraiu-se os dados da planilha operacional fornecida pela empresa. Esta planilha contém registros de tarefas e movimentações executadas durante um dia específico de operação. Utilizou-se filtros para isolar somente as tarefas de *picking*, selecionando um conjunto específico de 31 tarefas de separação para os experimentos. Organizou-se os dados na ordem cronológica em que as tarefas foram realizadas, bem como a sequência dos endereços visitados em cada uma delas. A Figura 22 apresenta a estrutura com parte dos dados utilizados para realizar as simulações.

Figura 22 – Estrutura dos dados utilizados para realizar as simulações

	A	B
1	Tarefa	Endereços
2	563978	04-40-1-304
3		05-25-1-302
4		05-31-1-302
5		10-41-1-101
6		10-9-1-101
7		11-27-1-101
8	563993	06-36-1-104
9		06-34-1-104
10		06-9-1-302
11		08-19-1-404
12		09-10-1-101
13		09-37-1-101
14	564005	04-25-1-104
15		04-25-1-104
16		05-2-1-102
17		05-41-1-304
18		06-36-1-101
19		08-11-1-101
20	564960	06-26-1-301
21		07-7-1-301
22		08-20-1-204

Fonte: O Autor (2024)

4.2.1 Cenário de Coletas Independentes: Otimização de Rotas Únicas

Este cenário descreve o método empregado pela empresa para a operação de separação de pedidos, ilustrando uma das formas potenciais de aplicação da ferramenta de otimização

desenvolvida. Nele, o operador inicia sua tarefa no ponto de partida, segue uma sequência ordenada de pontos de coleta e, após alcançar o último ponto, retorna ao ponto inicial. Para os experimentos, o ponto de coleta 118 corresponde ao depósito (ver Figura 51).

Esse processo é análogo à estrutura observada no problema do caixeiro viajante (Seção 2.4.2.1), visto que, assim como no caixeiro viajante, busca-se a rota mais eficiente para visitar um conjunto de locais (pontos de coleta) de maneira a minimizar a distância total percorrida (roteiro de separação), retornando ao ponto de origem (área de depósito) após percorrer todos os locais do caminho (endereços de coleta).

Diante disso, neste cenário, adotou-se uma estratégia de otimização baseada no PCV, na qual todas as tarefas de coleta iniciam e terminam no depósito. Portanto, o operador e o robô, ambos, começam e terminam suas operações nesta área de depósito. A solução para o problema consiste em construir uma rota que permita coletar todos os produtos e retornar ao depósito, com o objetivo de minimizar a distância percorrida ao longo dessa rota.

4.2.2 Cenário de Coletas Encadeadas: Otimização de Rotas Sequenciais

Neste cenário, adota-se um processo alternativo para a execução das tarefas de separação, eliminando a necessidade do operador retornar à área de depósito. Após finalizar uma tarefa de *picking*, o robô encaminha-se automaticamente de volta à área de depósito, permitindo que o operador permaneça aguardando um novo robô vazio no local do último ponto de coleta da rota que acabou de ser realizada. A seguir, com a atribuição de uma nova tarefa de separação, o operador desloca-se até o início da próxima rota de separação, onde dá continuidade ao seu trabalho de coleta normalmente.

Uma vez que as tarefas de separação são disponibilizadas de maneira sequencial, é importante destacar que a otimização é executada individualmente para cada uma, pois o método adotado não contempla a antecipação de tarefas futuras. Isso significa que o processo de otimização está ciente apenas da última tarefa que foi otimizada e da tarefa corrente em processo de otimização. Essa abordagem decorre das limitações do WMS atualmente utilizado pela empresa, bem como da intenção de que a ferramenta de otimização não altere os fluxos operacionais existentes. Devido a essa restrição, mesmo que disponível a informação de mais de uma tarefa de separação, não é possível alterar ou mesclar os endereços de coleta que as compõem. Desta forma, para este cenário, foi empregada a abordagem de otimização do PCVA.

Dessa forma, todas as rotas de coleta podem começar no depósito ou em qualquer outro ponto previamente mapeado no armazém. A primeira rota sempre inicia no depósito, enquanto as demais podem partir de qualquer local e terminar em outro ponto, conforme necessário. A solução para o problema envolve a determinação de uma rota ideal para realizar a coleta de todos os produtos sem a obrigatoriedade de retornar ao ponto de origem, permitindo o início e a conclusão da rota em qualquer ponto do armazém. Essa abordagem tem o objetivo de minimizar a distância percorrida pelo operador humano e eliminar deslocamentos desnecessários de ida e

volta ao depósito.

4.2.3 Margem de Otimização

A fim de possibilitar uma avaliação mais adequada dos ganhos obtidos com as abordagens de otimização, foi utilizada uma medida de diferença percentual em relação à menor distância possível de ser obtida na realização de uma rota de separação. Esse método de comparação é particularmente importante ao se considerar a abordagem PCVA, em que as estratégias de otimização podem exigir a definição prévia de pontos de início e/ou fim da rota, afetando o resultado final.

Para permitir essa avaliação, eliminando o viés introduzido pelas especificidades de cada estratégia de otimização avaliada, utilizou-se o modelo de otimização baseado no PCVA sem impor restrições de início ou fim. Essa abordagem possibilitou identificar a rota mais eficiente para cada tarefa de separação considerada na análise, resultando na menor distância possível de ser percorrida. Os resultados obtidos, incluindo as distâncias mínimas individuais para cada tarefa de separação e a distância total mínima, são apresentados parcialmente na Tabela 6.

Tabela 6 – Distâncias mínimas possíveis para cada tarefa de *picking*

<i>Picking</i>	Distância mínima possível (m)
563981	208,00
563977	84,07
563962	74,53
563934	58,07
564026	117,00
...	...
564066	52,00
564006	76,27
564024	76,27
564076	73,67
564080	47,67
Total	2.766,42

4.2.4 Resultados

Nesta seção, são apresentados os detalhes dos experimentos realizados e dos resultados obtidos com base nos dois cenários de avaliação definidos, a partir do emprego das abordagens de otimização PCV e PCVA, e suas estratégias de uso. Em todos os cenários, foram calculadas as rotas para a sequência completa de 31 tarefas de separação.

4.2.5 Resultados do cenário de coletas independentes (PCV)

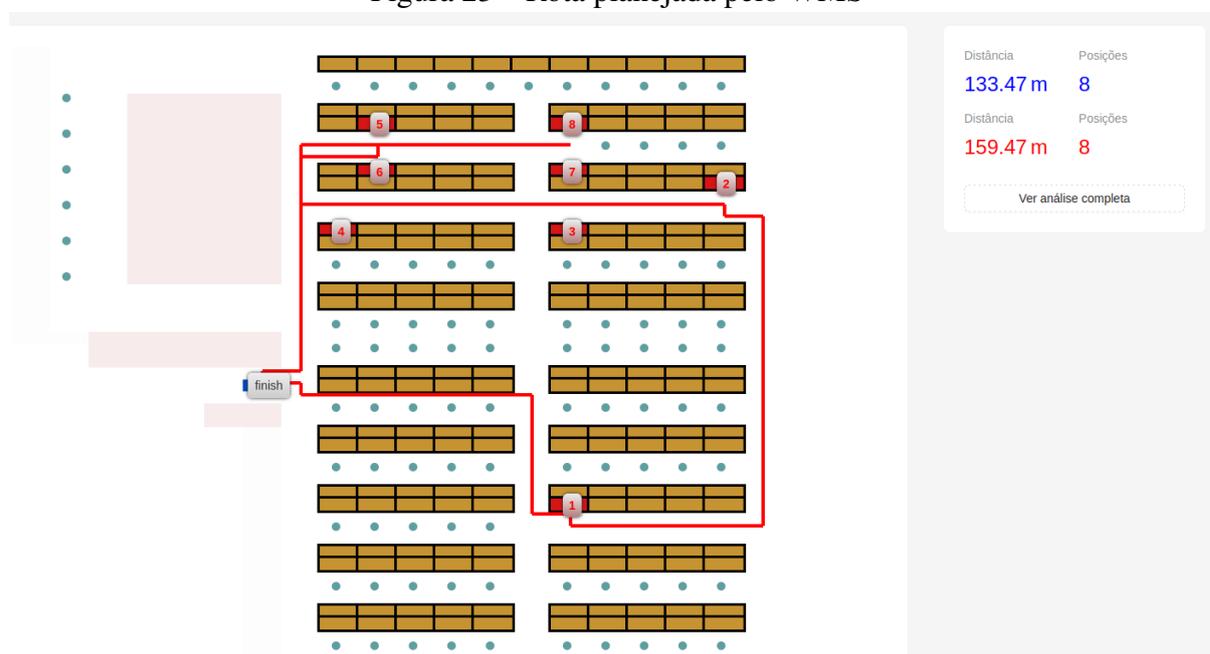
Neste cenário, tanto o operador quanto o robô iniciam e finalizam seu deslocamento na área de depósito, seguindo um processo alinhado com aquele utilizado pela empresa. A distância planejada para essas atividades é estabelecida com base em uma sequência predeterminada de pontos de coleta, conforme indicado na planilha operacional. Essa planilha é criada levando em conta a política de sequenciamento especificada pelo WMS. Além disso, à distância calculada, adiciona-se o trajeto desde o ponto do depósito até o primeiro ponto de coleta da tarefa e, da mesma forma, do último ponto do trajeto de volta ao ponto do depósito. Essa abordagem facilita a comparação com os resultados proporcionados pelo otimizador. A distância que foi planejada pelo WMS para executar as 31 tarefas é detalhada na Tabela 7.

Tabela 7 – Resultados consolidados do WMS para coletas independentes

Qtd. <i>Pickings</i>	Distância Planejada (metros)
31	4.884,53

A Figura 23 apresenta um exemplo de rota que foi planejada pelo WMS e executada por um operador da empresa.

Figura 23 – Rota planejada pelo WMS



Fonte: O Autor (2024)

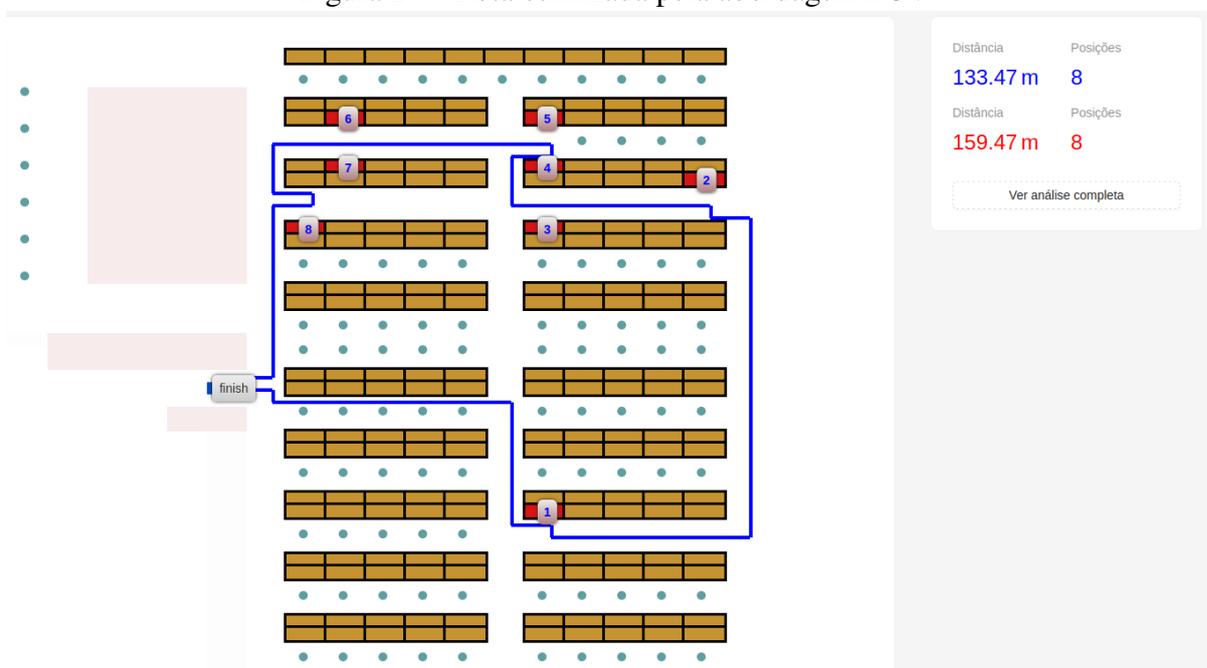
A Tabela 8 apresenta os resultados consolidados obtidos pela aplicação da otimização das rotas da lista de 31 tarefas selecionadas.

Tabela 8 – Comparação de Distâncias (em metros) Planejadas e Otimizadas

Qtd. <i>Pickings</i>	Dist. Planejada	Dist. Otimizada	Dist. Reduzida	Otimização Média (%)
31	4.884,53	4.099,33	785,20	16,08

A Figura 24 mostra a mesma tarefa da Figura 23, porém agora com a rota otimizada. Neste caso, é possível observar uma redução de 26 metros, ou 16,30%, da distância percorrida quando comparadas a rota planejada e a otimizada.

Figura 24 – Rota otimizada pela abordagem PCV



Fonte: O Autor (2024)

Os resultados apresentados anteriormente consideram que tanto o robô quanto o operador humano retornam ao depósito após cada tarefa de coleta. No entanto, para reduzir o deslocamento do operador humano, é possível avaliar um cenário alternativo em que apenas o robô retorna ao depósito, enquanto o operador permanece na área de *picking* (esse cenário é melhor explorado na próxima Seção). Para a tarefa de *picking* seguinte, o robô se desloca do depósito até o primeiro ponto da nova coleta, enquanto o operador humano parte do último ponto da tarefa anterior para o primeiro ponto da tarefa subsequente. Com isso, é possível observar uma redução na distância percorrida pelo operador. Para realizar essa análise, a Tabela 9 apresenta uma comparação entre os deslocamentos realizados pelos operadores humanos no cenário planejado (WMS) e no otimizado. Os dados foram divididos em quatro métricas:

- Com Retorno: Distâncias percorridas quando o operador acompanha o robô tanto para iniciar quanto para voltar ao depósito;
- Sem Retorno: Distâncias percorridas quando o operador permanece na área de *picking*;
- Durante *Picking*: Distâncias percorridas dentro de cada tarefa de *picking*;
- Entre *Pickings*: Distância necessária para se deslocar entre o final de uma tarefa de *picking* e o início da próxima. No cenário com retorno, inclui o deslocamento do depósito até o primeiro ponto de coleta, bem como o retorno do último ponto de

coleta ao depósito. No cenário sem retorno, corresponde à distância entre o último ponto da tarefa anterior e o primeiro ponto da tarefa seguinte.

Tabela 9 – Comparação das distâncias percorridas pelos operadores humanos

Distâncias do Operador	Planejado WMS (m)	Otimizado (m)	Redução (m)	Melhoria (%)
Com Retorno				
Entre <i>Pickings</i>	1.765,40	1.094,78	670,62	37,99
Durante <i>Picking</i>	3.119,13	3.004,55	114,58	3,54
Total	4.884,53	4.099,33	785,20	16,08
Sem Retorno				
Entre <i>Pickings</i>	1.162,20	626,60	535,60	46,09
Durante <i>Picking</i>	3.119,13	3.004,55	114,58	3,67
Total	4.281,33	3.631,15	650,18	15,19

Ao comparar as estratégias com retorno e sem retorno, observa-se que, no caso planejado, o operador percorre 603,20 metros a menos, o que corresponde a uma redução de 12,31% no deslocamento. Já no cenário otimizado, a diferença no deslocamento do operador é de 468,18 metros a menos, representando uma redução de 11,42%.

A Tabela 9 destaca também como a eliminação do retorno ao depósito pode reduzir o deslocamento dos operadores, independentemente do nível de otimização aplicado. Além disso, ao analisar os resultados que evidenciam os deslocamentos dos operadores fora da tarefa de *picking* (linha “Entre *Pickings*”) na categoria “Com Retorno”, fica claro a elevada distância de deslocamentos improdutivos que precisam ser realizados. Por fim, os resultados demonstram que o processo otimizado apresenta melhores resultados em todos os cenários analisados, com valores de deslocamento totais menores em comparação ao planejamento do WMS.

4.2.6 Resultados do cenário de coletas encadeadas (PCVA)

Neste cenário foi simulado um processo de operação contínuo. Para isso, tomou-se por base a sequência de execução das 31 tarefas de *picking* que foram extraídas da planilha operacional. A primeira coleta tem origem no ponto correspondente ao depósito, e as demais tarefas de separação seguem conforme planejado pelo WMS, sem que haja retorno para o depósito, com exceção da última tarefa.

Os resultados obtidos a partir das simulações permitiram comparar as rotas planejadas pelo WMS e as rotas geradas pela ferramenta de otimização, quando realizadas a mesma sequência de tarefas de coleta. Além disso, foram calculadas as distâncias de deslocamento entre o último ponto visitado de uma tarefa de *picking* e o primeiro ponto da próxima tarefa, bem como as distâncias adicionais relativas ao deslocamento inicial (ida) e final (retorno) do robô em relação ao depósito.

A Tabela 10 apresenta as distâncias consolidadas calculadas com base na política de sequenciamento adotada pelo WMS, considerando o operador humano durante a execução das tarefas. Esta tabela destaca:

- a) Quantidade de tarefas de *picking*: refere-se à quantidade de tarefas de coleta analisadas;
- b) Distância de deslocamento entre *pickings*: refere-se ao deslocamento necessário para o operador se deslocar do último ponto visitado de uma tarefa de coleta para o primeiro ponto da tarefa subsequente;
- c) Distância durante *picking*: corresponde ao deslocamento realizado pelo operador dentro de cada tarefa para coletar os itens planejados;
- d) Distância total percorrida: engloba tanto as distâncias percorridas durante as tarefas quanto as distâncias adicionais entre elas.

Tabela 10 – Resumo das distâncias percorridas pelo operador humano - WMS

Quantidade de Tarefas de <i>Picking</i>	Distância entre <i>Pickings</i> (m)	Distância durante <i>Pickings</i> (m)	Distância Total Percorrida (m)
31	603,17	3.114,82	3.717,99

Adicionalmente, a Tabela 11 apresenta os deslocamentos relacionados ao robô no mesmo cenário de referência. Estes valores incluem as distâncias iniciais e finais necessárias para iniciar e concluir a sequência de tarefas, além da distância total percorrida. Os deslocamentos considerados são:

- a) Distância inicial (depósito ao 1º ponto): Representa a distância percorrida pelo robô do depósito até o primeiro ponto de coleta;
- b) Distância final (retorno ao depósito): Refere-se à distância do último ponto de coleta até o depósito;
- c) Distância adicional total (inicial + final): Corresponde à soma das distâncias inicial e final;
- d) Distância total percorrida: Soma da distância adicional (inicial + final) e das distâncias internas para realizar as tarefas de *picking*, obtidos da tabela anterior (Distância durante as *pickings*).

Tabela 11 – Distâncias percorridas pelo robô - WMS

Distância Inicial (depósito ao 1º ponto)	Distância Final (retorno ao depósito)	Distância Adicional (inicial + final)	Distância Total Percorrida
984,90	874,13	1.859,03	4.973,85

Essas métricas permitiram avaliar o esforço envolvido nas rotas planejadas pelo WMS. Além disso, elas servem de base para comparar com as estratégias de otimização propostas. A seguir, serão apresentadas as quatro estratégias desenvolvidas.

4.2.6.1 Partida Ajustada: Ponto Próximo (Estratégia 1)

Nesta estratégia, foram identificadas as rotas para a sequência completa de 31 tarefas de separação, iniciando a primeira *picking* pelo ponto correspondente ao depósito, conforme ilustrado na Figura 25a. A sequência prossegue de maneira encadeada, conforme determinado pela ferramenta de otimização, utilizando o método do PCVA para identificar a rota mais eficiente, deixando o destino final das rotas em aberto. As tarefas subsequentes, como por exemplo a segunda tarefa de coleta, mostrada na Figura 25b, consideram o último ponto visitado da coleta anterior, para identificar o ponto de coleta subsequente mais próximo que integra a tarefa seguinte. Este ponto então se torna o novo ponto de partida para a otimização da rota em questão. Esse procedimento foi repetido sucessivamente até a realização da última tarefa de coleta, que foi finalizada no depósito. As Figuras 25a, 25b e 25c mostram as rotas de coleta das três primeiras tarefas submetidas à estratégia de otimização 1.

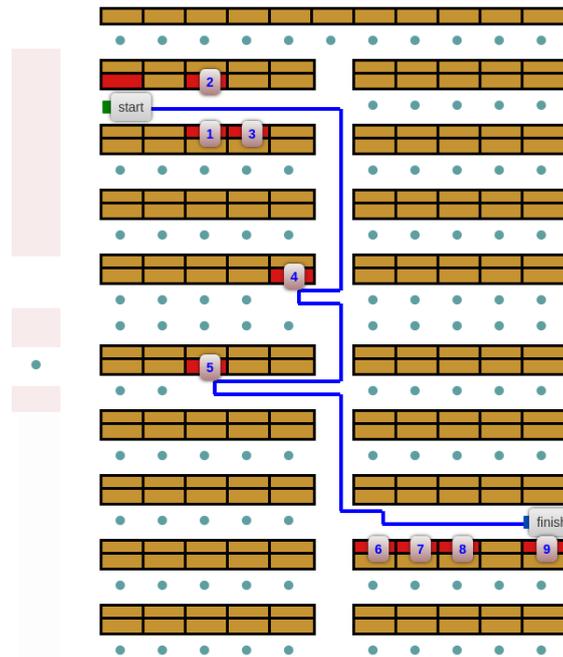
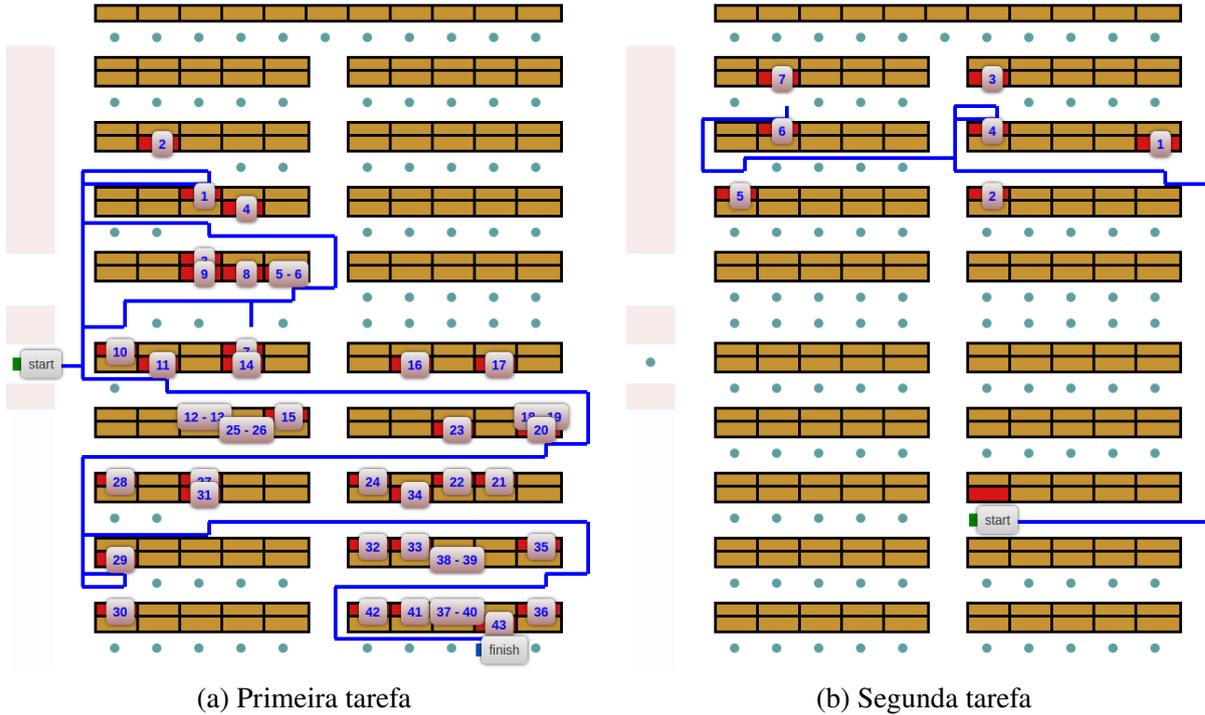
A Tabela 12 consolida as métricas de distâncias percorridas pelo operador humano e pelo robô no contexto da Estratégia 1. É apresentada uma comparação direta entre os resultados dessa otimização e do planejamento original realizado pelo WMS, destacando as reduções obtidas e os percentuais de melhoria alcançados.

Tabela 12 – Comparação com Planejamento WMS - Estratégia 1

Distâncias	WMS (m)	Estratégia 1 (m)	Redução (m)	Melhoria (%)
Operador Humano				
Entre tarefas	603,17	440,26	162,91	27,01
Durante tarefas	3.114,82	2.894,68	220,14	7,07
Total	3.717,99	3.334,94	383,05	10,31
Robô				
Inicial	984,90	775,67	209,23	21,25
Final	874,13	973,27	-99,14	-11,35
Adicional total	1.859,03	1.748,93	110,10	5,92
Total	4.973,85	4.643,61	330,24	6,64

A Figura 26 exibe um gráfico que apresenta a margem de otimização alcançada em cada uma das tarefas. O cálculo da margem de otimização é efetuado com base nos valores de distância mínima possível apresentados na Tabela 6 e representa, percentualmente, o quanto foi possível atingir da distância mínima possível.

Figura 25 – Rotas otimizadas a partir da Estratégia 1

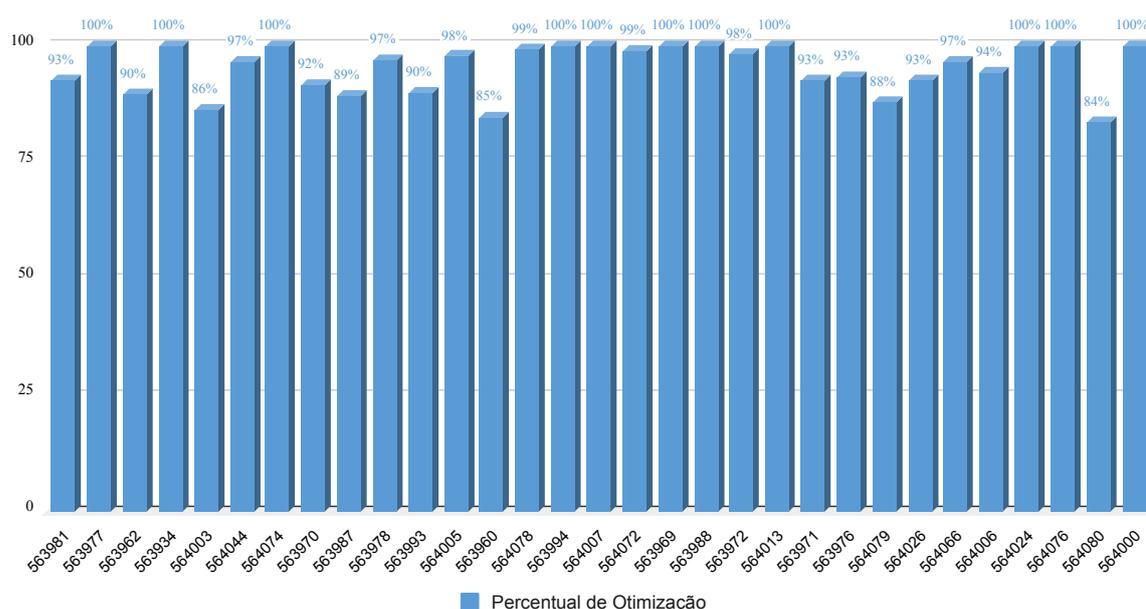


Fonte: O Autor (2024)

4.2.6.2 Partida Ajustada: Integra o Último Ponto à Rota Atual (Estratégia 2)

Esta estratégia adota a mesma base lógica que a Estratégia 1, introduzindo, no entanto, uma mudança na maneira de como o primeiro ponto de coleta de uma próxima tarefa de *picking* é selecionado para a otimização. Aqui, o ponto final de uma tarefa de coleta foi utilizado como ponto de partida na otimização da rota seguinte. Essa abordagem permitiu que o otimizador

Figura 26 – Nível de otimização alcançado em cada tarefa - Estratégia 1



Fonte: O Autor (2024)

determinasse a rota mais eficiente, considerando esse ponto específico como origem.

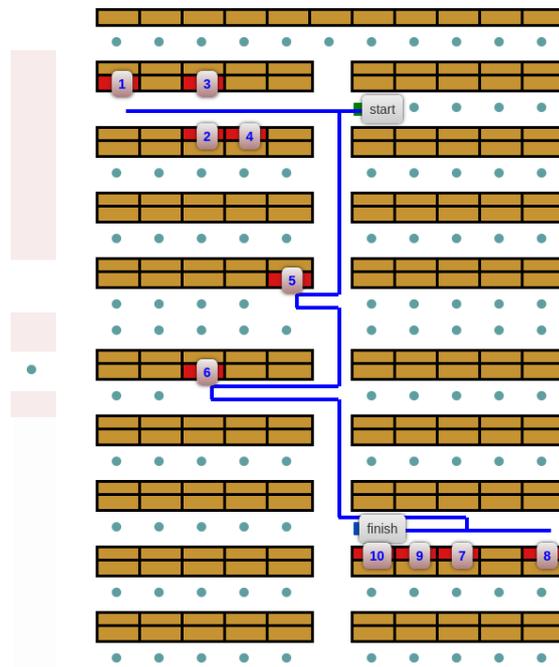
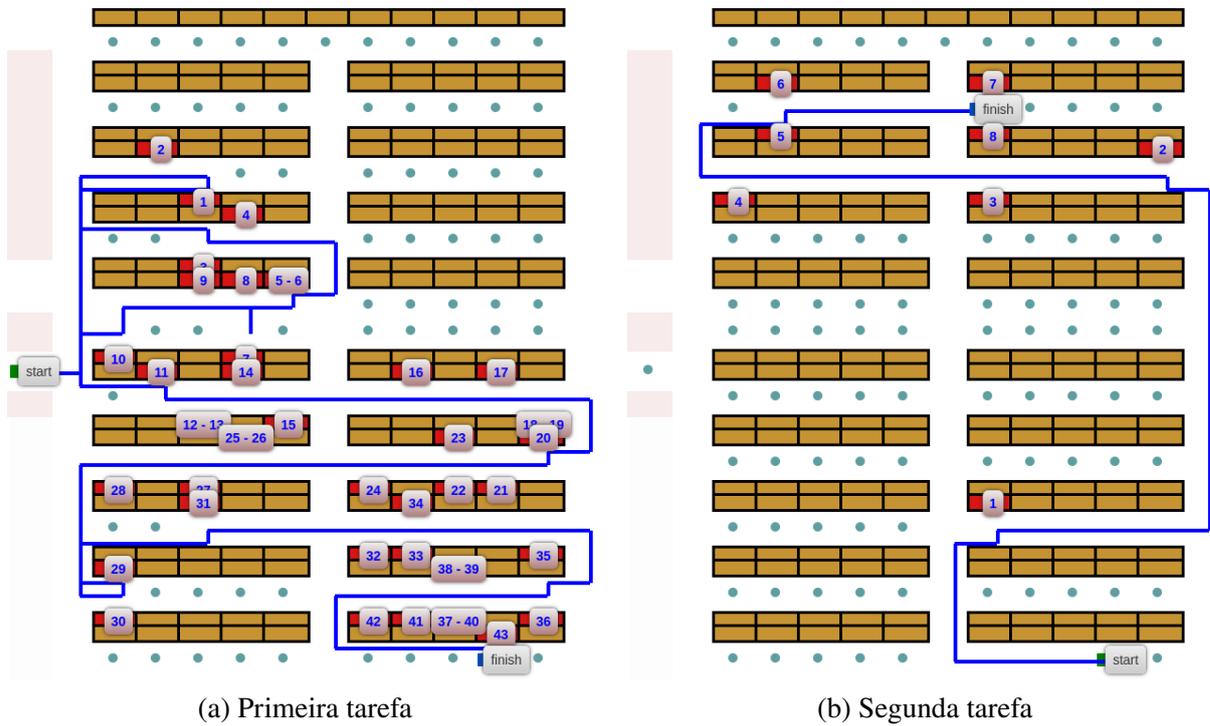
Logo após concluir a otimização, o referido ponto foi excluído da rota para que os cálculos de distância do trajeto fossem feitos corretamente. A distância percorrida a partir desse ponto até o primeiro ponto efetivo de coleta da tarefa seguinte foi então considerada como medida de deslocamento entre as tarefas de separação. As Figuras 27a, 27b e 27c mostram o comportamento das três primeiras tarefas de separação submetidas à Estratégia 2.

A Tabela 13 consolida as métricas de distâncias percorridas pelo operador humano e pelo robô no contexto da Estratégia 2. É apresentada uma comparação direta entre os resultados dessa otimização e do planejamento original realizado pelo WMS, destacando as reduções obtidas e os percentuais de melhoria alcançados.

Tabela 13 – Comparação com Planejamento WMS - Estratégia 2

Distâncias	WMS (m)	Estratégia 2 (m)	Redução (m)	Melhoria (%)
Operador Humano				
Entre tarefas	603,17	446,30	156,87	26,01
Durante tarefas	3.114,82	2.814,10	300,72	9,65
Total	3.717,99	3.260,40	457,59	12,31
Robô				
Inicial	984,90	755,73	229,17	23,27
Final	874,13	994,93	-120,80	-13,82
Adicional total	1.859,03	1.750,67	108,36	5,83
Total	4.973,85	4.564,77	409,08	8,22

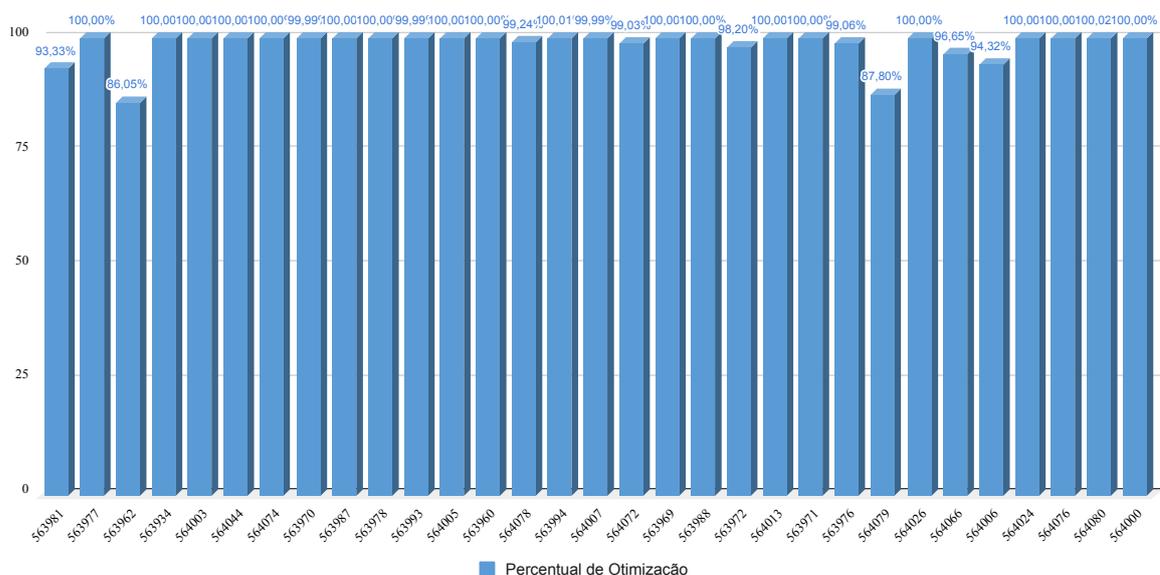
Figura 27 – Rotas otimizadas a partir da Estratégia 2



Fonte: O Autor (2024)

A Figura 28 exibe um gráfico que apresenta a margem de otimização alcançada em cada uma das tarefas. O cálculo da margem de otimização é efetuado com base nos valores de distância mínima possível apresentados na Tabela 6 e representa, percentualmente, o quanto foi possível atingir da distância mínima possível.

Figura 28 – Nível de otimização alcançado em cada tarefa - Estratégia 2



Fonte: O Autor (2024)

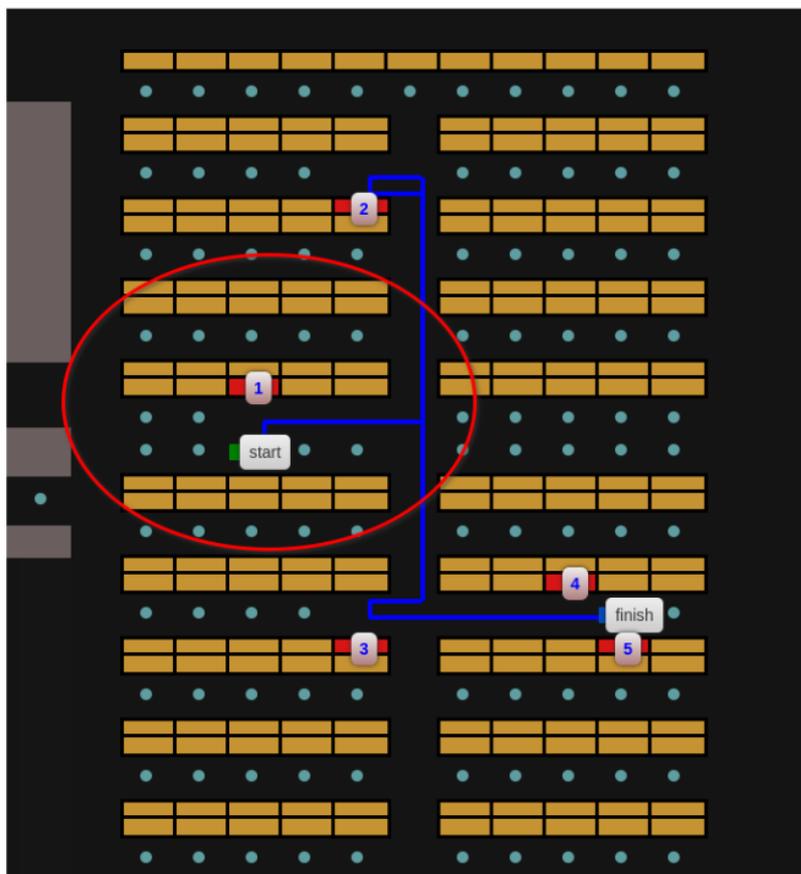
4.2.6.3 Adaptativa: Seleção Estratégica de Pontos Extremos (Estratégia 3)

Na estratégia 3, a primeira tarefa de separação teve como origem o depósito e foi definido um ponto “extremo” como ponto final. Ponto extremo se refere aos endereços que ficam tanto na extremidade inferior (próximo ao corredor 3 - ver Figura 51) como superior (próximo ao corredor 12 - ver Figura 51) do armazém. A segunda tarefa de separação (até a antepenúltima), utilizou como base o último ponto visitado em uma tarefa para encontrar o ponto mais próximo da tarefa seguinte para definir como início do caminho. Após esta definição, foi identificado o ponto mais extremo (do lado oposto) para definir como ponto final de coleta. Nos casos em que havia mais de um ponto extremo, foi definido aquele mais próximo ao corredor do depósito, por serem os endereços que armazenavam itens com maior curva de classificação, devido à sua proximidade com a área de depósito. Na última tarefa, o destino foi definido como o depósito.

A escolha de configurar o ponto inicial e final como “extremos” teve como objetivo evitar que a abordagem PCVA resultasse em rotas que terminassem em endereços próximos da região central do armazém, mesmo que isso levasse a rotas específicas mais curtas. Isso se justificou pelo fato de que, nas tarefas subsequentes, era altamente provável que o operador precisasse fazer uma coleta na metade superior e, em seguida, se deslocar para a metade inferior. Portanto, fazer esse percurso iniciando por um ponto central do armazém fazia com que o operador tivesse um deslocamento maior, já que ele tinha que subir e depois descer, ou vice e versa. A Figura 29 representa a situação aqui descrita.

A Tabela 14 consolida as métricas de distâncias percorridas pelo operador humano e pelo robô no contexto da Estratégia 3. Ela apresenta uma comparação direta entre os resultados dessa otimização e do planejamento original realizado pelo WMS, destacando as reduções obtidas e os percentuais de melhoria alcançados.

Figura 29 – Rota iniciando por um ponto de coleta central do armazém



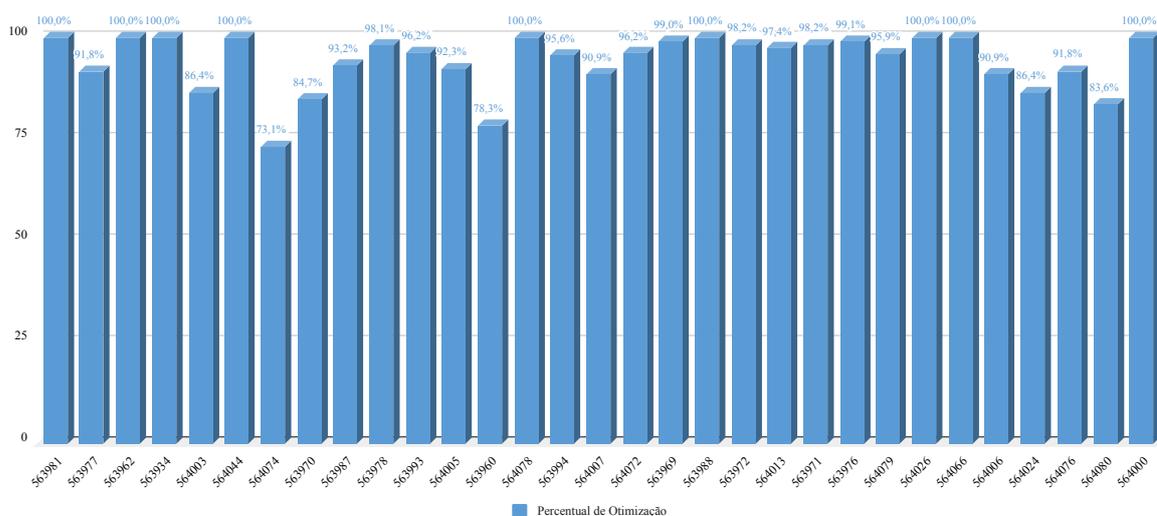
Fonte: O Autor (2024)

Tabela 14 – Comparação com Planejamento WMS - Estratégia 3

Distâncias	WMS (m)	Estratégia 3 (m)	Redução (m)	Melhoria (%)
Operador Humano				
Entre tarefas	603,17	428,12	175,05	29,02
Durante tarefas	3.114,82	2.899,86	214,96	6,90
Total	3.717,99	3.327,98	390,01	10,49
Robô				
Inicial	984,90	895,27	89,63	9,10
Final	874,13	905,67	-31,54	-3,61
Adicional total	1.859,03	1.800,93	58,10	3,12
Total	4.973,85	4.700,79	273,06	5,49

A Figura 30 exibe um gráfico que apresenta a margem de otimização alcançada em cada uma das tarefas. O cálculo da margem de otimização é efetuado com base nos valores de distância mínima possível apresentados na Tabela 6 e representa, percentualmente, o quanto foi possível atingir da distância mínima possível.

Figura 30 – Nível de otimização alcançado em cada tarefa - Estratégia 3



Fonte: O Autor (2024)

4.2.6.4 Combinação Estratégica: Extremos e Ajuste da Partida (Estratégia 4)

Esta estratégia toma por base a ideia aplicada na Estratégia 3, que define o ponto final de uma tarefa de coleta como sendo um dos extremos do armazém, e a combina com a Estratégia 2. Nesse caso, o ponto final de uma tarefa de coleta foi utilizado como ponto de partida para a otimização da rota da tarefa seguinte. Essa abordagem permitiu que o otimizador identificasse a rota mais eficiente, considerando esse ponto específico como origem.

Após a otimização, o ponto final da coleta de referência foi removido da rota e a distância percorrida a partir dele até o primeiro ponto efetivo de coleta da tarefa seguinte foi então utilizado como medida de deslocamento entre as tarefas de separação.

Tabela 15 – Comparação com Planejamento WMS - Estratégia 4

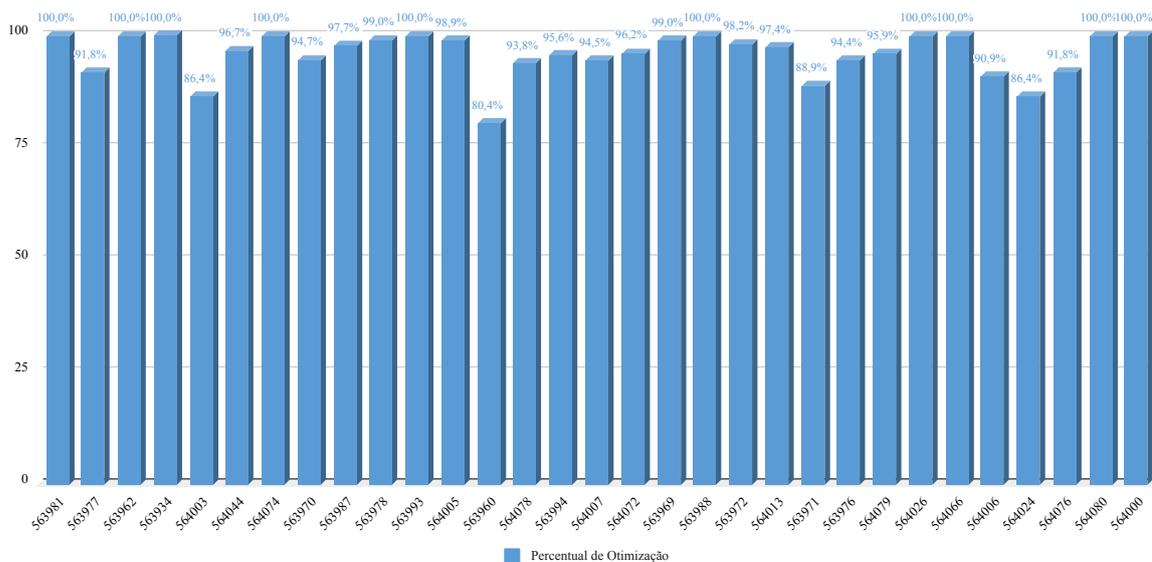
Distâncias	WMS (m)	Estratégia 4 (m)	Redução (m)	Melhoria (%)
Operador Humano				
Entre tarefas	603,17	370,06	233,11	38,65
Durante tarefas	3.114,82	2.872,99	241,83	7,76
Total	3.717,99	3.243,05	474,94	12,77
Robô				
Inicial	984,90	878,80	106,10	10,77
Final	874,13	938,60	-64,47	-7,38
Adicional total	1.859,03	1.817,40	41,63	2,24
Total	4.973,85	4.690,39	283,46	5,70

A Tabela 15 consolida as métricas de distâncias percorridas pelo operador humano e pelo robô no contexto da Estratégia 4. Ela apresenta uma comparação direta entre os resultados dessa

otimização e do planejamento original realizado pelo WMS, destacando as reduções obtidas e os percentuais de melhoria alcançados.

A Figura 31 exibe um gráfico que apresenta a margem de otimização alcançada em cada uma das tarefas. O cálculo da margem de otimização é efetuado com base nos valores de distância mínima possível apresentados na Tabela 6 e representa, percentualmente, o quanto foi possível atingir da distância mínima possível.

Figura 31 – Nível de otimização alcançado em cada tarefa - Estratégia 4



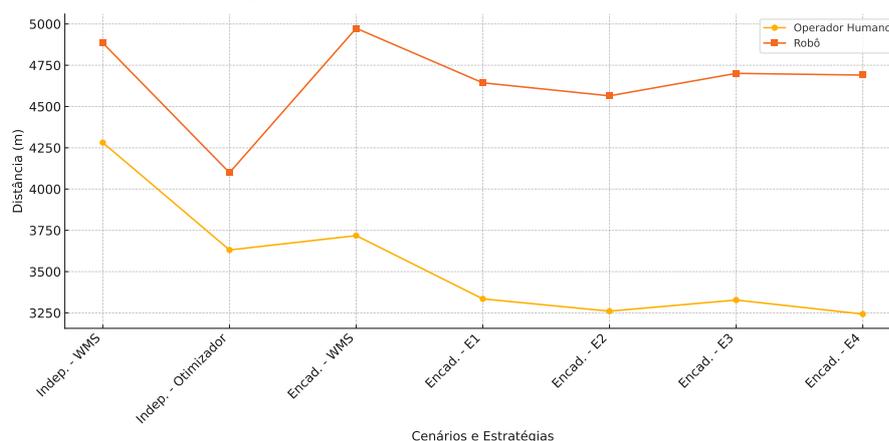
Fonte: O Autor (2024)

4.3 ANÁLISE E RECOMENDAÇÕES

Para analisar os resultados apresentados, é importante focar na comparação dos cenários de otimização considerados e seu desempenho em relação ao método tradicional utilizado pelo WMS. As análises tomaram por base dois cenários principais: coletas independentes e coletas encadeadas, usando modelos baseados no PCV e sua adaptação PCVA.

Com base na análise da otimização das coletas independentes, pode-se afirmar que esta abordagem adota uma estratégia voltada para reduzir os deslocamentos do robô. Isso ocorre porque a otimização é baseada na definição de rotas que sempre têm início e fim no depósito, reduzindo a distância percorrida pelo robô em cada tarefa. No entanto, a escolha de focar na otimização das rotas do robô (Independentes - PCV) tem um impacto direto no operador humano, que tem suas distâncias percorridas aumentadas. De forma inversa, quando a otimização prioriza as rotas do operador humano (Encadeadas - PCVA), a distância total percorrida pelo robô aumenta. Esse comportamento, que demonstra o impacto de priorizar um objetivo em detrimento do outro, pode ser observado no gráfico da Figura 32, na qual os cenários que priorizam a otimização do robô são identificados com o prefixo “Índep.”, enquanto os cenários que favorecem o operador humano são “Encad.”.

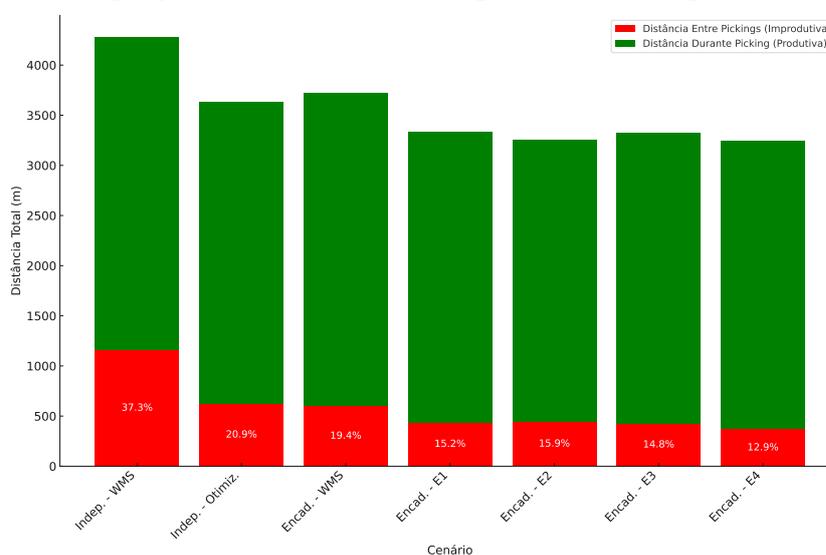
Figura 32 – Comparação de Distâncias Totais (Operador vs. Robô)



Fonte: O Autor (2024)

Com base na Figura 33, é possível identificar a proporção de deslocamentos improdutivos em cada cenário ¹. Nos cenários otimizados com as estratégias encadeadas (E1, E2, E3 e E4), observa-se uma redução desses deslocamentos. Em particular, ao comparar o cenário encadeado planejado pelo WMS com a estratégia de otimização E4, destaca-se que a distância improdutiva foi reduzida de 603,17 m (WMS) para 370,06 m, representando uma redução de 38,65%.

Figura 33 – Proporção de deslocamentos improdutivos dos operadores humanos



Fonte: O Autor (2024)

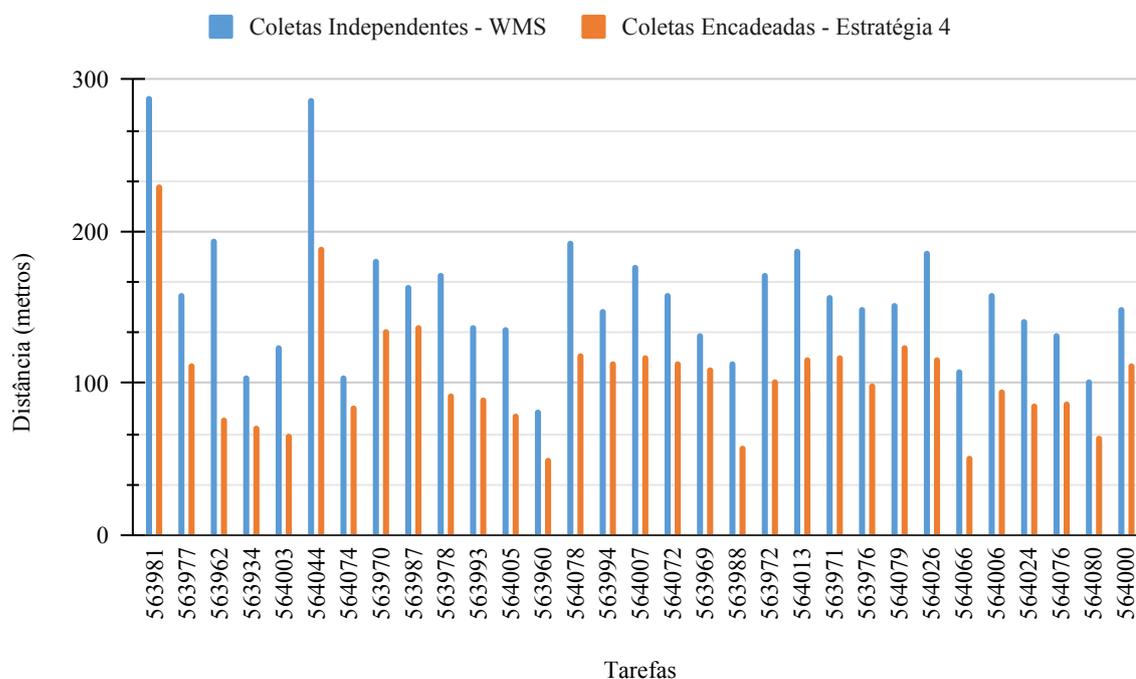
Com base nos resultados apresentados, o primeiro aspecto a ser destacado é o fato do cenário encadeado permitir uma maior redução na distância percorrida pelo operador, se comparado com cenário de coletas independentes.

Ao comparar o cenário de coletas independentes planejado pelo WMS com o cenário de coletas encadeadas otimizado pela Estratégia 4, observou-se uma redução na distância percorrida

¹ Considera-se deslocamento improdutivo, no cenário independente, o trajeto do depósito ao primeiro ponto de coleta e o retorno do último ponto de coleta ao depósito. Já no cenário encadeado, corresponde à distância entre o último ponto da tarefa anterior e o primeiro ponto da tarefa seguinte.

de 1.641,48 metros, o que representou uma diminuição de 33,61% no percurso total. A Figura 34 mostra um gráfico comparativo entre as distâncias de deslocamento realizadas pelo operador para atender cada uma das tarefas seguindo esses dois cenários de coletas.

Figura 34 – Gráfico comparativo entre as distâncias de deslocamento do operador no cenário de coletas independentes e no cenário de coletas encadeadas.



Fonte: O Autor (2024)

No cenário de coletas encadeadas, as Estratégias 1 e 2 mostraram melhorias consistentes na redução da distância total, com ganhos de 10,31% e 12,31%, respectivamente. A Estratégia 2, que considerou na otimização o último ponto da coleta anterior como início para determinação da próxima tarefa de coleta, se mostrou ligeiramente mais eficaz que a Estratégia 1.

A Estratégia 3, que opta por iniciar e terminar as coletas em pontos extremos, também demonstrou uma melhora, com ganhos de 10,49%. Indicando que posicionar o fim das tarefas em pontos estratégicos pode reduzir o deslocamento subsequente.

A Estratégia 4, que combina as abordagens das Estratégias 2 e 3, resultou em uma redução no deslocamento total do operador de 12,77%. Isso sugere que a combinação de iniciar a tarefa no último ponto da rota anterior e selecionar pontos extremos como destinos finais pode ser uma abordagem mais eficiente. Ao examinar os cenários de coletas encadeadas, fica evidente que a Estratégia 4 apresentou a melhor otimização em relação ao planejamento realizado pelo WMS, promovendo uma redução total de 474,94 metros.

Embora a redução de deslocamentos tenha alcançado uma melhoria de 12,77%, os resultados não foram tão expressivos devido a maior parte das tarefas de separação envolver uma quantidade pequena de posições de coleta (em torno de 5 pontos de coleta), fazendo com que,

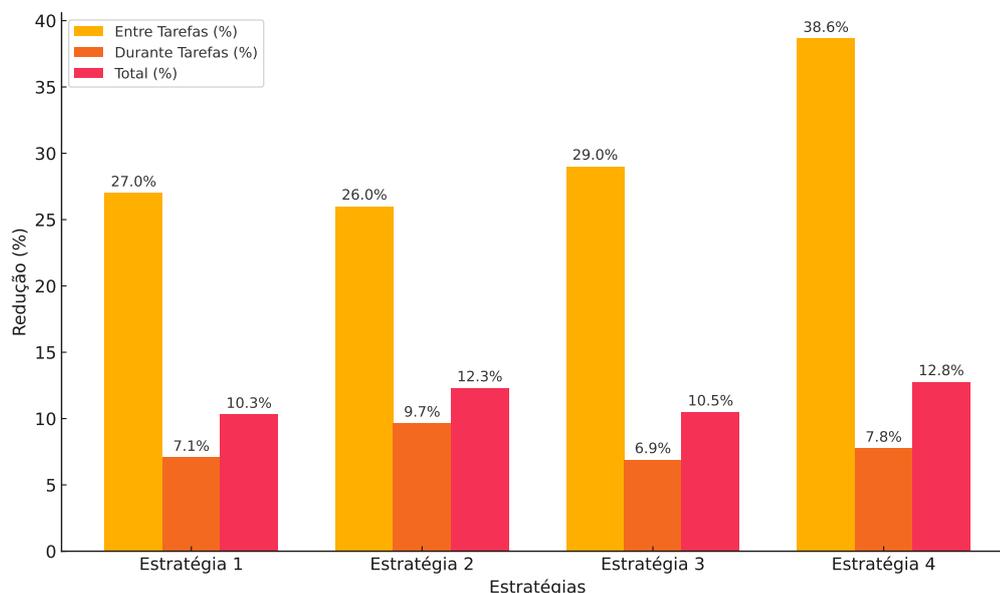
mesmo com a aplicação de uma lógica simples de roteamento, como por exemplo, aquela empregada pelo WMS, permitiu alcançar uma margem alta de eficácia da otimização, de 87,25%.

Tabela 16 – Comparação de Estratégias - Distância Durante Picking

Estratégia	Dist. Estratégia (m)	Dist. Mínima (m)	Diferença (m)	GAP (%)
Estratégia 1	2.894,68	2.766,42	128,26	4,64
Estratégia 2	2.814,10	2.766,42	47,68	1,71
Estratégia 3	2.899,86	2.766,42	133,44	4,83
Estratégia 4	2.872,99	2.766,42	106,57	3,85

A Tabela 16 apresenta uma comparação dos resultados obtidos com as quatro estratégias propostas para a execução das 31 tarefas de coleta avaliadas neste estudo. São apresentadas nesta tabela, as distâncias percorridas pelo operador em cada estratégia, a distância mínima possível, a diferença de distância para alcançar o mínimo possível e a sua representação percentual (GAP). Os percentuais de proximidade variaram de 1,71% (Estratégia 2) a 4,83% (Estratégia 3), evidenciando a eficácia geral das abordagens propostas.

Figura 35 – Comparação de desempenhos relativos de cada estratégia



Fonte: O Autor (2024)

O gráfico da Figura 35 apresenta as reduções percentuais de deslocamento “Entre Tarefas”, “Durante Tarefas” e “Total” do operador humano em cada estratégia, quando comparadas com aquelas geradas pelo emprego da abordagem do WMS. Observa-se no gráfico uma relação entre a priorização em reduzir a distância entre *pickings* e reduzir a distância durante o *picking*, indicando que focar exclusivamente em uma métrica pode impactar negativamente a outra.

Quando comparada à otimização total, é essencial considerar não apenas a redução durante o *picking*, mas também a redução da distância entre *pickings*. Estratégias que priorizam

exclusivamente uma dessas métricas podem impactar negativamente o desempenho geral. Por exemplo, a Estratégia 4, apesar de apresentar apenas a segunda melhor redução da distância durante o *picking* (7,80%), apresentou os melhores resultados dentre as estratégias avaliadas, pois alcançou a maior redução na distância entre *pickings* (38,65%). Esse equilíbrio evidencia a importância de considerar ambas as métricas para contribuir para a eficiência global do processo.

Considerando a aplicação da Estratégia 4, que apresentou a maior redução na distância percorrida pelo operador entre as estratégias avaliadas, essa otimização diária pode resultar em uma economia de tempo e esforço se considerado um período prolongado. Observa-se também que os ganhos obtidos só seriam alcançados se as condições de operação fossem as mesmas que aquelas consideradas nos experimentos, sem interferências externas que podem impactar na operação. Assumindo então essa condição, pode-se considerar que a economia de 12,77% gerada por esta estratégia permite realizar as tarefas em menos tempo, a cada 8 dias de trabalho, aproximadamente, sendo possível ganhar o equivalente a um dia inteiro de operação. Projetando esse ganho ao longo de um ano, com aproximadamente 250 dias úteis, a empresa poderia reduzir o tempo operacional em cerca de 31 dias no período avaliado, o que equivale a aproximadamente 6 semanas de operação. No entanto, para confirmar essa projeção e assegurar que os resultados são consistentes em diferentes cenários operacionais, seria necessário realizar mais experimentos.

Com base na comparação entre o cenário de coletas independentes planejado pelo WMS e a Estratégia 4, foi possível determinar os resultados operacionais observados após a introdução da coleta colaborativa humano-robô. Os resultados demonstram uma redução de 33,61% no deslocamento dos operadores, indicando uma otimização. Assumindo novamente uma condição favorável, conforme destacado anteriormente, isso equivale a ganhar um dia adicional de trabalho a cada 3 dias de operação. Projetando essa otimização ao longo de um ano, a economia alcançaria aproximadamente 121 dias úteis adicionais, correspondendo a cerca de 24 semanas de trabalho.

Com base nos resultados obtidos, é possível concluir que as estratégias de otimização sugeridas apresentam ganhos em termos de redução de deslocamentos improdutivos. Essa redução impacta diretamente na eficiência operacional e na diminuição do esforço físico dos operadores. No entanto, essa análise também revela que ao focar na redução dos deslocamentos do operador, os deslocamentos necessários para o robô atender a demanda, aumentam.

Dessa forma, a escolha da melhor estratégia depende de uma avaliação dos objetivos da operação e dos recursos disponíveis. Cabe ao decisor equilibrar os benefícios para os operadores e os custos operacionais dos robôs, considerando as prioridades específicas da operação.

5 CONCLUSÃO

O presente estudo se concentrou no desenvolvimento de uma ferramenta computacional para solução do problema de roteamento de único separador em um contexto de colaboração humano-robô em armazéns que seguem o método tradicional de coleta *picker-to-parts*. Este problema é relevante, pois, a coleta de pedidos representa uma das tarefas operacionais mais custosa de um armazém, devido à alta demanda por trabalho manual e os grandes deslocamentos, e além disso, a aquisição de recursos de automação não garante, necessariamente, melhorias significativas se não for corretamente explorada para uma implementação adequada.

A metodologia adotada, baseada na modelagem computacional, no agrupamento de endereços em pontos de coleta estratégicos e no uso de técnicas de otimização, possibilitou a avaliação comparativa das estratégias propostas para otimização das rotas de coleta. A partir dos experimentos realizados, foi possível observar que algumas estratégias podem trazer melhorias reduzindo os deslocamentos improdutivos, com destaque para aquelas que buscam minimizar a necessidade de retorno ao depósito por parte dos robôs. Essa abordagem, ao permitir que os operadores permaneçam nas regiões de *picking*, sem a necessidade de retornos constantes ao depósito, demonstraram uma redução de 33,61% no deslocamento dos operadores, quando comparada com o método tradicional de *picking* utilizado pela empresa.

Após a definição da estratégia operacional para o uso colaborativo entre humanos e robôs, que visa a eliminar parte dos deslocamentos improdutivos, os resultados da comparação entre as rotas planejadas pelo sistema WMS atual e as rotas otimizadas de acordo com a nova proposta operacional, indicaram que as rotas geradas pela aplicação da Estratégia 4 da ferramenta de otimização foram, em média, 12,77% mais curtas. A aplicação dessa estratégia de otimização alcançou 96,15% de eficácia global, apresentando um gap de 3,85% para atingir a máxima otimização possível de cada rota.

Os resultados obtidos demonstraram a eficácia da ferramenta desenvolvida na melhoria das rotas de coleta, ao comparar as soluções otimizadas com as estratégias atualmente utilizadas pela empresa. A metodologia de otimização de rotas adotada, fundamentada em modelos como o Problema do Caixeiro Viajante (PCV) e o Problema do Caixeiro Viajante Aberto (PCVA), empregando abordagens exatas e heurísticas aplicadas de forma dinâmica conforme o tamanho das instâncias, mostrou-se eficiente e aplicável em um contexto real, provando ser uma estratégia viável para resolver o problema em questão.

Contudo, ao confrontar as distâncias percorridas pelo operador humano e pelo robô, identificaram-se ganhos e considerações que devem ser avaliadas. A redução do deslocamento do operador diminui o desgaste físico e melhora a eficiência operacional, mas também pode aumentar as distâncias percorridas pelos robôs, implicando em maior necessidade de recargas de bateria, riscos de espera e ajustes no dimensionamento da frota. Assim, cabe ao decisor avaliar esses aspectos conflitantes para definir a estratégia mais adequada ao contexto da operação.

Além disso, algumas limitações foram observadas. A ferramenta desenvolvida não considera fatores como o dimensionamento das frotas de robôs, a capacidade de carga dos mesmos, ou a variações nos *layouts* dos CDs, que podem influenciar diretamente os resultados obtidos. Além disso, o número de amostras e experimentações realizadas foi limitado, o que pode ter influenciado nos resultados encontrados. Experimentos adicionais com um maior número de cenários e variações no *layout* dos armazéns poderiam proporcionar uma avaliação mais robusta da eficácia das abordagens testadas, ajudando a validar ainda mais os resultados.

A pesquisa ofereceu contribuições relevantes para o estudo de otimização em operações logísticas, com foco especial no contexto de colaboração humano-robô. Futuros estudos podem expandir a abordagem adotada, incluindo a integração de múltiplos operadores e robôs, levando em consideração restrições como a capacidade de carga e o dimensionamento da frota de robôs. Além disso, seria importante considerar o cenário completo de tarefas de coleta, com o objetivo de reduzir os deslocamentos entre diferentes tarefas, melhorando ainda mais a eficiência operacional. Outro ponto relevante seria expandir a área de análise para incluir outros fatores que possam impactar diretamente o desempenho das rotas, como o *layout* do armazém e as estratégias de armazenamento dos produtos, com a intenção de aproximar itens frequentemente coletados juntos, reduzindo o tempo total de deslocamento.

REFERÊNCIAS

- AARTS, E.; AARTS, E. H.; LENSTRA, J. K. **Local Search in Combinatorial Optimization**. [S.l.]: Princeton University Press, 2003.
- AZADEH, K.; KOSTER, R. D.; ROY, D. Robotized and automated warehouse systems: Review and recent developments. **Transportation Science**, v. 53, 06 2019.
- BALAJI, K.; KUMAR, V. S. Multicriteria inventory abc classification in an automobile rubber components manufacturing industry. **Procedia CIRP**, v. 17, p. 463–468, 2014. ISSN 2212-8271.
- BALAS, E.; NG, S. On the set covering polytope: Ii. lifting the facets with coefficients in $\{0, 1, 2\}$. **Mathematical Programming**, v. 45, p. 1–20, 08 1989.
- BIANCHI, L. et al. A survey on metaheuristics for stochastic combinatorial optimization. **Natural Computing**, v. 8, n. 2, p. 239–287, Jun 2009. Disponível em: <<https://doi.org/10.1007/s11047-008-9098-4>>.
- BLOSS, R. Simultaneous sensing of location and mapping for autonomous robots. **Sensor Review**, Emerald Group Publishing Limited, v. 28, n. 2, p. 102–107, 2008. ISSN 0260-2288. Disponível em: <<https://doi.org/10.1108/02602280810856651>>.
- BOFFEY, T. B. The travelling salesman and chinese postman problems. In: _____. **Graph Theory in Operations Research**. London: Macmillan Education UK, 1982. p. 148–172. ISBN 978-1-349-16675-6. Disponível em: <https://doi.org/10.1007/978-1-349-16675-6_7>.
- BOYSEN, N.; de Koster, R.; WEIDINGER, F. Warehousing in the e-commerce era: A survey. **European Journal of Operational Research**, v. 277, n. 2, p. 396–411, 2019. ISSN 0377-2217.
- BOYWITZ, D.; SCHWERDFEGER, S.; BOYSEN, N. Sequencing of picking orders to facilitate the replenishment of a-frame systems. **IIE Transactions**, v. 51, p. 1–27, 08 2018.
- BRUCKER, P. **Scheduling Algorithms**. 5th ed. ed. Springer, 2007. ISBN 9783540695158; 354069515X. Disponível em: <<https://ftp.idu.ac.id/wp-content/uploads/ebook/ip/BUKU%20SCHEDULING/Scheduling%20Algorithms.pdf>>.
- CARVALHO, J. C. de. **Logística e gestão da cadeia de abastecimento**. 2. ed. [S.l.]: Edições Sílabo, 2017. ISBN 978-972-618-894-0.
- CAUCHICK-MIGUEL, P. **Metodologia Científica para Engenharia**. Elsevier Editora Ltda., 2019. ISBN 9788535290714. Disponível em: <<https://books.google.com.br/books?id=SN-LDwAAQBAJ>>.
- CHAN, F. T.; CHAN, H. Improving the productivity of order picking of a manual-pick and multi-level rack distribution warehouse through the implementation of class-based storage. **Expert Systems with Applications**, v. 38, n. 3, p. 2686–2700, 2011. ISSN 0957-4174.
- CHARLES, P. An evaluation of order picking routing policies,. **International Journal of Operations e Production Management**, v. 17, p. 1098–1111, 11 1997.
- CHRISTOFIDES, N. The shortest hamiltonian chain of a graph. **SIAM Journal on Applied Mathematics**, Society for Industrial and Applied Mathematics, v. 19, n. 4, p. 689–696, 1970. ISSN 00361399. Disponível em: <<http://www.jstor.org/stable/2100152>>.

CIČKOVÁ, Z.; BREZINA, I.; PEKÁR, J. Open traveling salesman problem with time windows. **1st Logistics International Conference**, Belgrade, Serbia, p. 40–43, November 2013.

COOK, W. J. et al. Combinatorial optimisation. **Wiley-Interscience Series in Discrete Mathematics and Optimization, USA**, Springer, v. 1, p. 998, 1998.

CORDENOSI, A. Z. **Ambientes, Objetos e Dialogicidade: uma estratégia de ensino superior em heurísticas e metaheurísticas**. Tese (Doutorado em Informática na Educação) — Universidade Federal do Rio Grande do Sul, Porto Alegre, 2008.

CORMEN, T. et al. **Introduction to Algorithms, third edition**. MIT Press, 2009. (The MIT Press). ISBN 9780262258104. Disponível em: <<https://books.google.com.br/books?id=F3anBQAAQBAJ>>.

CREPINSEK, M.; LIU, S.-H.; MERNIK, M. Exploration and exploitation in evolutionary algorithms: A survey. **ACM Comput. Surv.**, Association for Computing Machinery, New York, NY, USA, v. 45, n. 3, jul 2013. ISSN 0360-0300. Disponível em: <<https://doi.org/10.1145/2480741.2480752>>.

DALLARI, F.; MARCHET, G.; MELACINI, M. Design of order picking system. **The International Journal of Advanced Manufacturing Technology**, v. 42, n. 1, p. 1–12, May 2009. ISSN 1433-3015.

DANTZIG, G.; FULKERSON, R.; JOHNSON, S. Solution of a large-scale traveling-salesman problem. **Journal of the Operations Research Society of America**, INFORMS, v. 2, n. 4, p. 393–410, 1954. ISSN 00963984.

DANTZIG, G. B.; RAMSER, J. H. The truck dispatching problem. **Management Science**, v. 6, n. 1, p. 80–91, 1959. Disponível em: <<http://www.jstor.org/stable/2627477>>.

DIAS, L. et al. Application of the fuzzy logic for the development of autonomous robot with obstacles deviation. **International Journal of Control, Automation and Systems**, v. 16, 01 2018.

DIJKSTRA, E. W. A note on two problems in connexion with graphs. **Numerische Mathematik**, v. 1, p. 269–271, 1959. Disponível em: <<https://doi.org/10.1007/BF01386390>>.

FERNANDES, K. dos S. **Logística: fundamentos e processos**. [S.l.]: IESDE BRASIL SA, 2012.

FILHO, M. **Desenvolvimento e aplicação de algoritmos heurísticos ao problema de alocação de espaço físico em universidades**. Dissertação (Mestrado) — UNIVERSIDADE ESTADUAL DE MARINGÁ (RI-UEM), 2008. Disponível em: <<http://repositorio.uem.br:8080/jspui/handle/1/2504>>.

FILHO, P. de F. **Introdução à modelagem e simulação de sistemas: com aplicações em Arena**. Visual Books, 2008. ISBN 9788575022283. Disponível em: <<https://books.google.com.br/books?id=VT-zkQEACAAJ>>.

FRAGAPANE, G. et al. Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda. **European Journal of Operational Research**, v. 294, n. 2, p. 405–426, 2021. ISSN 0377-2217.

FRAZELLE, E. **World-class warehousing and material handling**. Second edition. [S.l.]: McGraw-Hill Education, 2016. ISBN 9780071842822; 0071842829; 9780071842839; 0071842837.

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**, v. 267, p. 1–15, 05 2018.

GOLDBARG, M. C.; LUNA, H. P. L. **Otimização combinatória e programação linear: modelos e algoritmos**. [S.l.]: Elsevier, 2005.

GOLDBERG, D. E.; HOLLAND, J. H. Genetic algorithms and machine learning. **Machine Learning**, v. 3, n. 2, p. 95–99, 1988. ISSN 1573-0565. Disponível em: <<https://doi.org/10.1023/A:1022602019183>>.

GOMORY, R. Outline of an algorithm for integer solutions to linear programs. **Bulletin of the American Mathematical Society**, v. 64, p. 275–278, 1958. Disponível em: <<http://dx.doi.org/10.1090/S0002-9904-1958-10224-4>>.

GOMORY, R. E. An algorithm for the mixed integer problem. In: . [S.l.: s.n.], 1960.

GROOVER, M. **Automation, Production Systems, and Computer-integrated Manufacturing**. Prentice Hall, 2008. (Pearson international edition). ISBN 9780132393218. Disponível em: <https://books.google.com.br/books?id=an-PTIoh_NoC>.

GROSSE, E.; GLOCK, C.; BALLESTER-RIPOLL, R. A simulated annealing approach for the joint order batching and order picker routing problem with weight restrictions. **International Journal of Operations and Quantitative Management**, v. 20, p. 65–83, 01 2014.

GUAN, M. Graphic programming using odd or even points. **Acta Mathematica Sinica**, p. 273–277, 1962.

GUE, K.; MELLER, R. Aisle configurations for unit-load warehouses. **IIE Transactions**, v. 41, p. 171–182, 03 2009.

GUE, K. R.; IVANOVIC, G.; MELLER, R. D. A unit-load warehouse with multiple pickup and deposit points and non-traditional aisles. **Transportation Research Part E: Logistics and Transportation Review**, Elsevier, v. 48, n. 4, p. 795–806, 2012.

GÓMEZ, A.; GALAFASSI, C.; CHIWIACOWSKY, L. A hybrid metaheuristic approach to solve the vehicle routing problem with time windows. In: **Proceedings of the 9th International Conference on Informatics in Control, Automation and Robotics (ICINCO-2012)**. SCITEPRESS (Science and Technology Publications, Lda.), 2012. p. 183–188. ISBN 978-989-8565-21-1. Disponível em: <<https://www.scitepress.org/Papers/2012/40300/40300.pdf>>.

HEIDELBERG, U. of. **TSPLIB**. 2023. Acessado: 12-04-2024. Disponível em: <<http://comopt.ifl.uni-heidelberg.de/software/TSPLIB95/tsp/>>.

HILLIER, F.; LIEBERMAN, G. **Introdução à Pesquisa Operacional**. AMGH, 2013. ISBN 9788580551198. Disponível em: <<https://books.google.com.br/books?id=-A88a0-KxQOC>>.

HOFFMAN, K.; RALPHS, T. Integer and combinatorial optimization. **Encyclopedia of Operations Research and Management Science**, 01 2013.

- HOLLOCKS, B. W. Guidelines for effective simulation in manufacturing management and control. **IFAC Proceedings Volumes**, v. 36, n. 3, p. 137–142, 2003. ISSN 1474-6670. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1474667017377479>>.
- HOMPEL, M.; SCHMIDT, T. Warehouse management: Automation and organisation of warehouse and order picking systems (intralogistik). **Fraunhofer IML**, 01 2007.
- HUSSAIN, K. et al. Metaheuristic research: a comprehensive survey. **Artificial Intelligence Review**, v. 52, n. 4, p. 2191–2233, 2019.
- JUAN, A. A. et al. A review of simheuristics: Extending metaheuristics to deal with stochastic combinatorial optimization problems. **Operations Research Perspectives**, v. 2, p. 62–72, 2015. ISSN 2214-7160. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S221471601500007X>>.
- KARASEK, J. An overview of warehouse optimization. **International Journal of Advances in Telecommunications, Electrotechnics, Signals and Systems**, v. 2, p. 111–117, 11 2013.
- KIRKPATRICK, S.; GELATT, C. D.; VECCHI, M. P. Optimization by simulated annealing. **Science**, v. 220, n. 4598, p. 671–680, 1983. Disponível em: <<https://www.science.org/doi/abs/10.1126/science.220.4598.671>>.
- KORTE, B.; VYGEN, J. **Combinatorial Optimization: Theory and algorithms**. 5. ed. Springer Berlin, Heidelberg, 2012. 660 p. (Algorithms and Combinatorics). ISBN 978-3-642-24488-9. Disponível em: <<https://doi.org/10.1007/978-3-642-24488-9>>.
- KOSTER, R. D.; DUC, T. L.; ROODBERGEN, K. J. Design and control of warehouse order picking: A literature review. **European Journal of Operational Research**, p. 481–501, 10 2007.
- KULAK, O.; ŞAHIN, Y.; TANER, M. Joint order batching and picker routing in single and multiple-cross-aisle warehouses using cluster-based tabu search algorithms. **Flexible Services and Manufacturing Journal - FLEX SERV MANUF J**, v. 24, 03 2012.
- KULKARNI, R. Association rule-extracting knowledge using market basket analysis. **Research Journal of Recent Sciences**, v. 1, 01 2012.
- LAWLER, E. L.; WOOD, D. E. Branch-and-bound methods: A survey. **Operations Research, INFORMS**, v. 14, n. 4, p. 699–719, 1966. ISSN 0030364X, 15265463.
- LE-ANH, T.; KOSTER, M. D. A review of design and control of automated guided vehicle systems. **European Journal of Operational Research**, v. 171, n. 1, p. 1–23, 2006. ISSN 0377-2217. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0377221705001840>>.
- LEE, M.-K.; ELSAYED, E. A. Optimization of warehouse storage capacity under a dedicated storage policy. **International Journal of Production Research**, Taylor and Francis, v. 43, n. 9, p. 1785–1805, 2005.
- LÖFFLER, M.; BOYSEN, N.; SCHNEIDER, M. Picker routing in agv-assisted order picking systems. **INFORMS Journal on Computing**, v. 34, 08 2021.
- MANZINI, R. **Warehousing in the Global Supply Chain**. [S.l.]: Springer, 2012. ISBN 978-1-4471-2273-9.

MATTABONI, P. J. **Autonomous mobile robot**. [S.l.]: Google Patents, 1987. US Patent 4,638,445.

MILLER, C. E.; TUCKER, A. W.; ZEMLIN, R. A. Integer programming formulation of traveling salesman problems. **Journal of the Association for Computing Machinery**, Association for Computing Machinery, v. 7, n. 4, p. 326–329, oct 1960. Disponível em: <<https://doi.org/10.1145/321043.321046>>.

MULLER, T. **Automated guided vehicles**. [S.l.]: Springer, 1983.

NEMHAUSER, G. L.; WOLSEY, L. A. **Integer and Combinatorial Optimization**. [S.l.]: John Wiley & Sons, 1988.

NORDIN, N.; MALAYSIA, U.; SERDANG, U. Heuristics and metaheuristics approaches for facility layout problems: a survey. 01 2016.

PANG, A.; CHAN, H. L. Data mining-based algorithm for storage location assignment in a randomised warehouse. **International Journal of Production Research**, v. 55, p. 1–18, 10 2016.

PAPADIMITRIOU, C.; STEIGLITZ, K. **Combinatorial Optimization: Algorithms and Complexity**. Dover Publications, 1998. (Dover Books on Computer Science). ISBN 9780486402581. Disponível em: <<https://books.google.com.br/books?id=cDY-joeCGoIC>>.

PAPADIMITRIOU, C. H. The euclidean travelling salesman problem is np-complete. **Theoretical Computer Science**, v. 4, n. 3, p. 237–244, 1977. ISSN 0304-3975. Disponível em: <<https://www.sciencedirect.com/science/article/pii/0304397577900123>>.

PEARL, J. **Heuristics: Intelligent Search Strategies for Computer Problem Solving**. [S.l.]: Addison-Wesley, 1984.

PERES, F.; CASTELLI, M. Combinatorial optimization problems and metaheuristics: Review, challenges, design, and development. **Applied Sciences**, v. 11, n. 14, 2021. ISSN 2076-3417. Disponível em: <<https://www.mdpi.com/2076-3417/11/14/6449>>.

PERRON, L.; FURNON, V. **OR-Tools**. 2024. Disponível em: <<https://developers.google.com/optimization/>>.

RABINER, L. Combinatorial optimization: algorithms and complexity. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, v. 32, n. 6, p. 1258–1259, 1984.

REEVES, C. **Modern Heuristic Techniques for Combinatorial Problems**. McGraw-Hill, 1995. (Advanced topics in computer science series). ISBN 9780077092399. Disponível em: <<https://books.google.com.br/books?id=oU95QgAACAAJ>>.

RENSBURG, L. Janse van. **Artificial intelligence for warehouse picking optimization-an NP-hard problem**. 2019.

RICHARDS, G. **Warehouse Management: A Complete Guide to Improving Efficiency and Minimizing Costs in the Modern Warehouse**. Kogan Page, 2018. ISBN 9780749479770. Disponível em: <<https://books.google.com.br/books?id=ysL1nAAACAAJ>>.

ROODBERGEN, K. J.; KOSTER, R. D. Routing order pickers in a warehouse with a middle aisle. **European Journal of Operational Research**, v. 133, p. 32–43, 08 2001a.

- ROODBERGEN, K. J.; KOSTER, R. de. Routing methods for warehouses with multiple cross aisles. **International Journal of Production Research**, Taylor e Francis, v. 39, n. 9, p. 1865–1883, 2001b.
- SACCHETIN, M. C. **Análise e implementação de algoritmos para localização e mapeamento de robôs móveis baseada em computação reconfigurável**. 120 p. Dissertação (Dissertação (Mestrado em Ciências de Computação e Matemática Computacional)) — Universidade de São Paulo, São Carlos, SP, 2005.
- SANTIS, R. D. et al. An adapted ant colony optimization algorithm for the minimization of the travel distance of pickers in manual warehouses. **European Journal of Operational Research**, Elsevier, v. 267, n. 1, p. 120–137, 2018.
- SILVA, M. T. d. **Hibridização de métodos exatos e heurísticos para a minimização do atraso ponderado no escalonamento de tarefas em máquinas paralelas**. 112 p. Dissertação (Mestrado em Informática) — Universidade Federal do Amazonas, Manaus, 2018.
- SILVA, T. A.; SILVA, G. O uso da metaheurística guided local search para resolver o problema de escala de motoristas de ônibus urbano. **TRANSPORTES**, v. 23, p. 105, 08 2015.
- SILVA, V. D.; ROCHE, J.; KONDOZ, A. Robust fusion of lidar and wide-angle camera data for autonomous mobile robots. **Sensors**, v. 18, n. 8, p. 2730, 2018. Disponível em: <<https://www.mdpi.com/1424-8220/18/8/2730>>.
- SILVER, E. An overview of heuristic solution methods. **Journal of The Operational Research Society - J OPER RES SOC**, v. 55, p. 936–956, 05 2004.
- SÖRENSEN, K.; SEVAUX, M.; GLOVER, F. A history of metaheuristics. **Handbook of Heuristics**, p. 791–808, 08 2018.
- SOUZA, M. J. F.; FILHO, N. M. **Programação de Horários em Escolas: Uma Aproximação por Metaheurísticas**. Tese (Doutorado) — Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2000.
- SSI SCHAEFER. **FTS 2Pick - Automated Guided Vehicles**. 2020. Acesso em: 11 de janeiro de 2024. Disponível em: <<https://www.ssi-schaefer.com/en-ae/products/conveying-transport/automated-guided-vehicles/fts-2pick-32648>>.
- TOMPKINS, J. et al. **Facilities Planning**. [S.l.]: Wiley, 2010. ISBN 9780470444047.
- TOTVS. **Nível de serviço logístico: entenda sua importância**. 2024. Acesso em: 10-04-2024. Disponível em: <<https://www.totvs.com/blog/gestao-logistica/nivel-de-servico-logistico/>>.
- VOUDOURIS, C. **Guided local search for combinatorial optimisation problems**. Tese (Doutorado) — Department of Computer Science, University of Essex, Colchester, 1997.
- VOUDOURIS, C.; TSANG, E. Guided local search and its application to the traveling salesman problem. **European Journal of Operational Research**, v. 113, n. 2, p. 469–499, 1999. ISSN 0377-2217. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S037722179800099X>>.
- VOUDOURIS, C.; TSANG, E. P. K. Guided local search. In: _____. **Handbook of Metaheuristics**. Boston, MA: Springer US, 2003. p. 185–218. ISBN 978-0-306-48056-0. Disponível em: <https://doi.org/10.1007/0-306-48056-5_7>.

WATERLOO, U. of. **TSPLIB**. 2023. Acessado: 12-04-2024. Disponível em: <<https://www.math.uwaterloo.ca/tsp/data/index.html>>.

WINSTON, W.; GOLDBERG, J. **Operations Research: Applications and Algorithms**. Thomson Brooks/Cole, 2004. (Operations Research: Applications and Algorithms). ISBN 9780534423629. Disponível em: <<https://books.google.com.br/books?id=tg5DAQAAIAAJ>>.

WOLSEY, L.; NEMHAUSER, G. **Integer and Combinatorial Optimization**. Wiley, 2014. (Wiley Series in Discrete Mathematics and Optimization). ISBN 9781118626863. Disponível em: <<https://books.google.com.br/books?id=MvBjBAAAQBAJ>>.

ZHOU, W. et al. Rfid-enabled flexible warehousing. **Decision Support Systems**, v. 98, p. 99–112, 2017. ISSN 0167-9236.

ZHU, J. Picking path optimization in dual zone type warehouse based on genetic algorithms. **International Journal of Economic Behavior and Organization**, v. 3, p. 41, 01 2015.

ŽULJ, I. et al. Order batching and batch sequencing in an amr-assisted picker-to-parts system. **European Journal of Operational Research**, v. 298, n. 1, p. 182–201, 2022. ISSN 0377-2217. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0377221721004616>>.

Apêndices

APÊNDICE A – ESTRATÉGIAS DE ARMAZENAMENTO

Segundo Koster, Duc e Roodbergen (2007), as estratégias de armazenamento são diretrizes que determinam onde os produtos recebidos devem ser armazenados (em quais posições da área de armazenamento), buscando equilibrar a máxima utilização do armazém, ao mesmo tempo em que visam a minimizar a distância percorrida pelos operadores durante a coleta.

Diversos fatores estão atrelados à disposição dos produtos no armazém, como a estratégia de coleta de pedidos, o *layout* de armazenamento, equipamentos de manuseio dos produtos, as características dos materiais, as tendências de demanda, sazonalidades e as taxas de rotatividade. A escolha adequada das estratégias de armazenamento contribuem significativamente para aprimorar o desempenho do processo de *picking* (CHAN; CHAN, 2011).

As duas principais estratégias de armazenamento são: i) aleatório/dinâmico (ou, caótico), ii) dedicado/fixo. No armazenamento aleatório, os produtos são armazenados nos locais disponíveis mais próximos após serem recebidos, sem uma designação prévia de um endereço específico a ser utilizado. Portanto, os produtos são colocados nos endereços vazios conforme a disponibilidade, sem uma lógica específica de organização, seja esta baseada em critérios de associação, frequência de pedidos ou similaridade de produtos. A desvantagem desta estratégia é que pode resultar em um aumento na distância percorrida pelos operadores encarregados pela separação de pedidos (KOSTER; DUC; ROODBERGEN, 2007).

O armazenamento fixo é uma estratégia na qual produtos específicos são atribuídos a endereços de armazenamento exclusivos. Cada SKU possui um endereço predeterminado e exclusivo para ele. Essa escolha é baseada em critérios específicos, como características do produto, demanda, tamanho, peso, classificação ou outros fatores (LEE; ELSAYED, 2005). Uma desvantagem dessa estratégia é que o local fixo permanece reservado mesmo para produtos que estão sem estoque físico no armazém, o que acaba resultando em uma menor utilização de espaço, visto que os endereços permanecerão vazios (MANZINI, 2012). No armazenamento fixo, alguns critérios usados para determinar o vínculo do produto com um endereço específico incluem:

- Posição com base na demanda: Produtos de alta demanda são armazenados em locais estratégicos, geralmente próximos da área de consolidação ou expedição. Já os produtos com baixa demanda são armazenados em endereços mais distantes (KOSTER; DUC; ROODBERGEN, 2007);
- Armazenamento com base na Classificação ABC: Os produtos e os endereços do armazém são categorizados em classes, e de acordo com a importância de cada produto, são designados para uma classe de endereços mais ou menos prioritária. A categorização dos produtos segue o Princípio de Pareto, ou seja, 80% das consequências advém de 20% das causas (BALAJI; KUMAR, 2014);

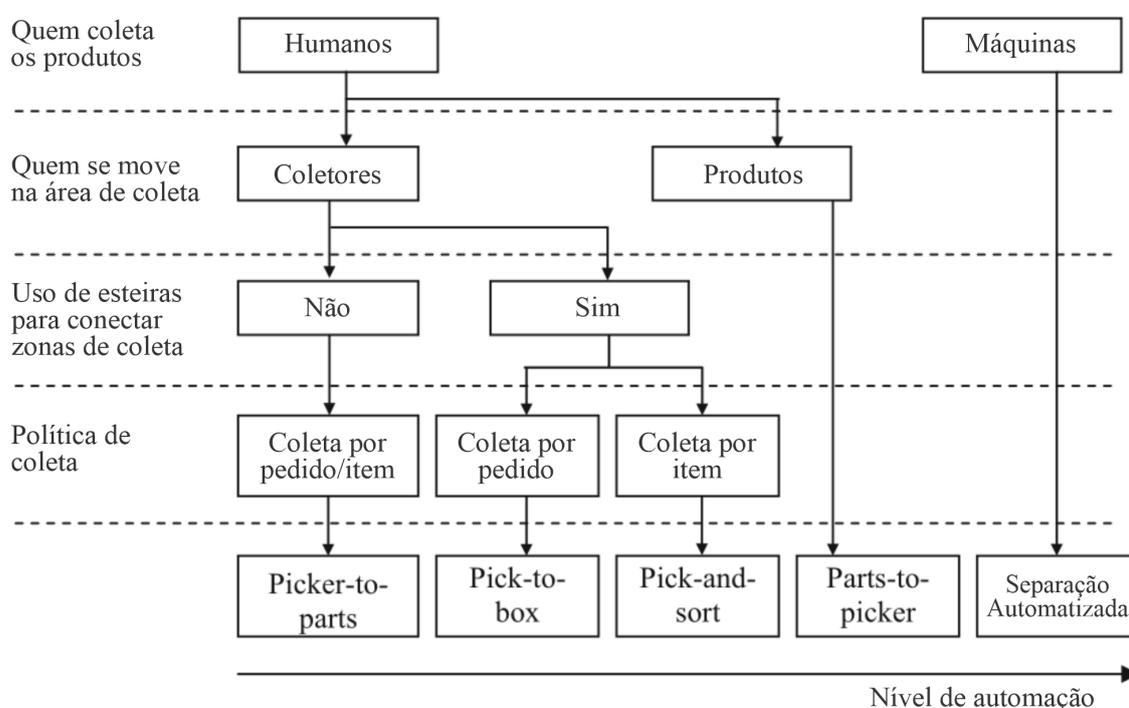
- Regras de associação: Definias a partir da análise de produtos que são frequentemente comprados juntos em pedidos antigos para prever associações entre itens em futuros pedidos (KULKARNI, 2012). Por exemplo, se os itens X e Y costumam ser comprados juntos com frequência, enquanto o item W está moderadamente associado a todos os outros itens X , Y e Z , e o item Z tem alguma relação com o item W , mas não muito com os itens X e Y , essa estratégia de armazenamento colocaria os itens X e Y próximos um do outro no estoque, o item W entre os itens X , Y e Z , e o item Z um pouco mais distante dos itens X e Y . Isso ajuda a usar melhor o espaço e reduzir a distância percorrida pelos operadores quando não há muitas variações nos pedidos ao longo do tempo (PANG; CHAN, 2016).

Por fim, com relação a essas estratégias, o uso de endereçamento fixo se mostra eficaz em ambientes com padrões previsíveis de pedidos, seguindo tendências consistentes ao longo do tempo. Por outro lado, em operações com pedidos menos previsíveis, recomenda-se a adoção do armazenamento dinâmico, por exemplo, em empresas de *e-commerce*, onde os pedidos dos clientes variam consideravelmente para uma ampla gama de SKUs (ZHOU et al., 2017).

APÊNDICE B – CATEGORIAS DE SISTEMAS DE COLETA

Existem diversas categorias de sistemas de coleta. Para categorizá-los, foi adotada a classificação proposta por Dallari, Marchet e Melacini (2009), a qual se baseia em quatro critérios distintos, permitindo a diferenciação entre cinco categorias de sistemas, conforme mostrado na Figura 36. Os critérios são: (i) quem é responsável pela coleta de produtos (humanos ou máquinas), (ii) quem se desloca na área de coleta (operadores ou produtos), (iii) é feito o uso de esteiras para movimentação dos materiais entre zonas de coleta e (iv) política/estratégia de coleta. Com base nesses critérios, as categorias resultantes são: *picker-to-parts*, *parts-to-picker*, *pick-to-box*, *pick-and-sort*, e separação totalmente automatizada.

Figura 36 – Categorização de sistemas de coleta de pedidos



Fonte: Adaptado de Dallari, Marchet e Melacini (2009).

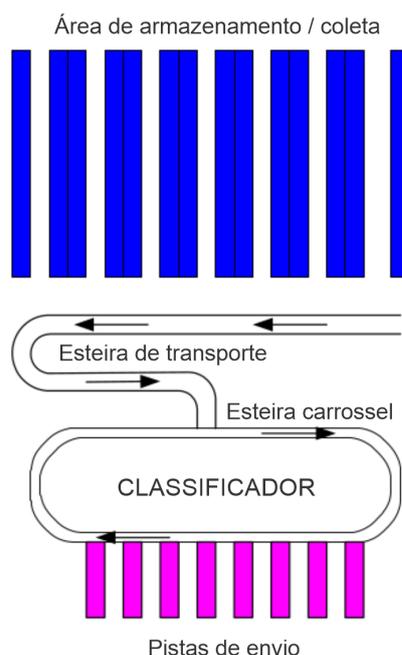
No sistema *picker-to-parts*, os operadores se deslocam até os endereços específicos dos produtos a serem coletados para atender os pedidos. Este é o método mais comum, geralmente são manuais e sem automação (KOSTER; DUC; ROODBERGEN, 2007).

Em sistemas *pick-to-box*, a área de coleta é subdividida em zonas conectadas por esteiras, e cada zona é designada a um ou mais operadores. Cada pedido de cliente é associado a uma caixa de coleta para então serem coletados de forma sequencial em cada zona. Após reunir todos os itens necessários na zona atual, a caixa de coleta é colocada na esteira e encaminhada para a próxima zona (DALLARI; MARCHET; MELACINI, 2009). Esse sistema geralmente funciona seguindo a estratégia de *picking* por zona discutida anteriormente.

Nos sistemas *pick-to-sort* (Figura 37), os operadores coletam os itens necessários para

atender a vários pedidos e os colocam em uma esteira operando em um circuito fechado que conecta a área de armazenagem dos produtos com a área de separação. Uma vez na esteira, um sistema computadorizado ligado a mecanismos automáticos de desvio determina o local de destino para cada item. Cada local de destino se refere a um pedido individual do cliente. Esses sistemas geralmente funcionam seguindo a estratégia de *picking* por ondas comentada anteriormente.

Figura 37 – Circuito de esteiras utilizado no sistema *pick-to-sort*



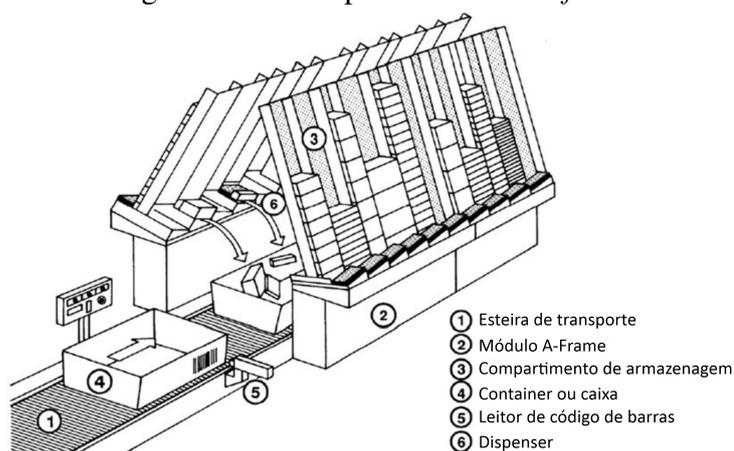
Fonte: Adaptado de Koster, Duc e Roodbergen (2007).

Nos sistemas *parts-to-picker*, os itens são transportados da área de armazenagem até estações de coleta (*picker*) por meio de máquinas automatizadas, onde os operadores aguardam para realizar a coleta. Nesse sistema, o operador é responsável apenas por selecionar o item e a quantidade requisitada. Após a coleta, os demais itens são devolvidos à área de armazenagem. Os benefícios desse método estão centrados na melhor utilização das horas de trabalho, uma vez que os operadores não precisam gastar tempo deslocando-se (DALLARI; MARCHET; MELACINI, 2009). Esses sistemas empregam tecnologias automatizadas, tais como carrosséis, Módulo de Elevação Vertical (VLM), *mini-loads*, Sistema de Recuperação e Armazenamento Automatizado (AS/RS), além dos AGVs e AMRs, detalhados na Seção 2.2.

Por fim, na categoria mais avançada tecnologicamente, a coleta dos produtos é realizada de maneira totalmente automatizada, sendo o sistema mais popular o *A-frame*, mostrado na Figura 38. Este sistema possui várias prateleiras ou compartimentos inclinados (formato de “A” quando visto de lado) contendo os produtos armazenados. Quando a caixa de coleta (associada a um pedido) que requer um determinado produto passa pelo endereço correspondente, os robôs (braços mecânicos) ou mecanismos automatizados (ventosas, esteiras, etc) são ativados. O A-

frame é capaz de processar tarefas de coleta com uma maior velocidade, resultando em uma redução no tempo necessário para a preparação de cada pedido. Mesmo com a coleta ocorrendo de forma totalmente automatizada, o *A-frame* ainda requer a intervenção de operadores humanos para realizar o armazenamento preciso dos produtos nos locais designados dentro de sua estrutura (AZADEH; KOSTER; ROY, 2019).

Figura 38 – Exemplo de sistema *A-frame*

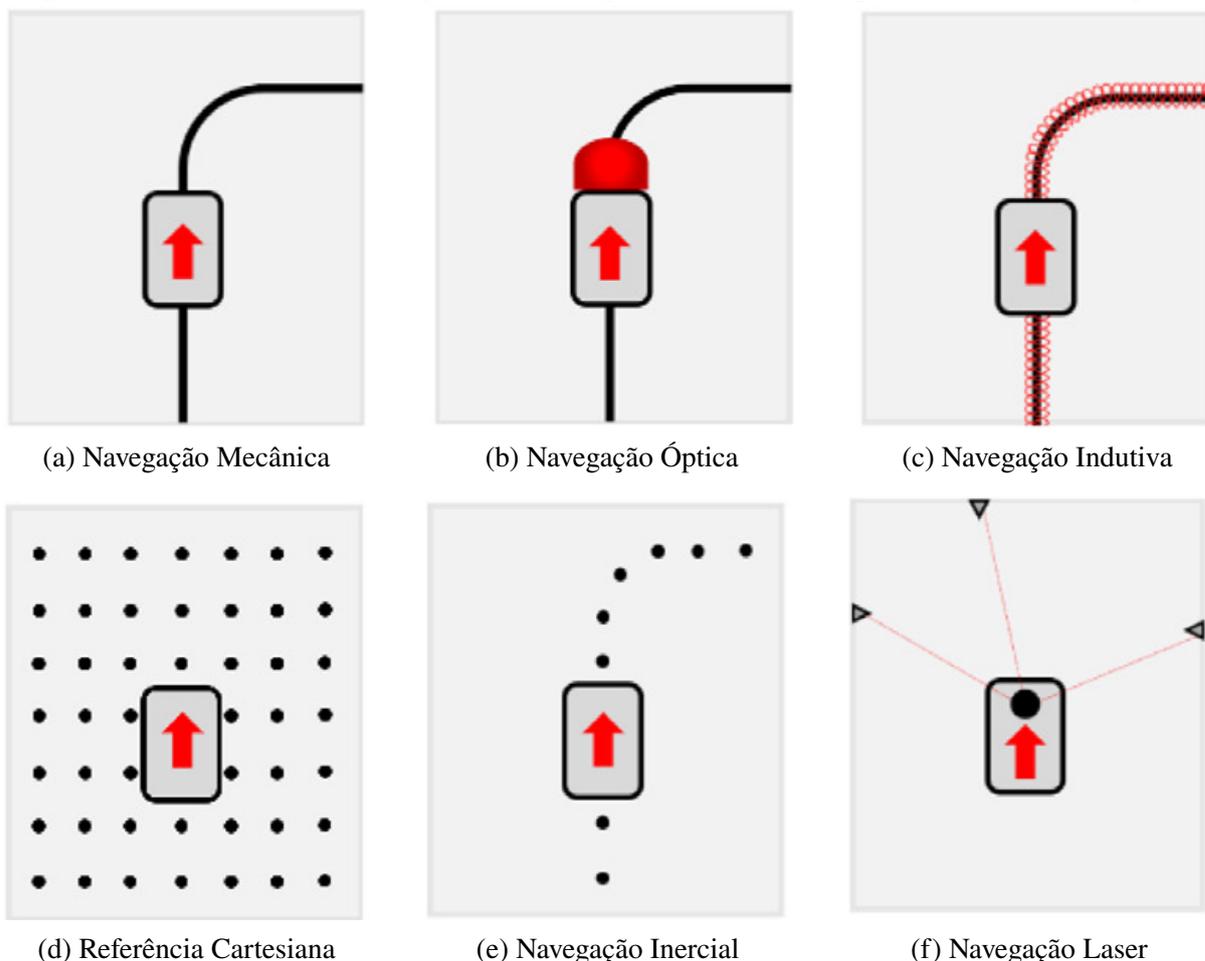


Fonte: Adaptado de Boywitz, Schwerdfeger e Boysen (2018).

APÊNDICE C – SISTEMAS DE ORIENTAÇÃO

O avanço na tecnologia de movimentação autônoma de materiais foi possível, em grande parte, pela inovação nos sistemas de orientação, que são essenciais para a navegação precisa desses veículos (GROOVER, 2008). A evolução desses sistemas, desde a navegação mecânica (Figura 39a) até a navegação baseada em visão (Figura 40), ilustra o avanço tecnológico que conferiu aos AMRs a capacidade de adaptarem-se de forma inteligente às mudanças no ambiente, diferenciando-os significativamente dos AGVs (FRAGAPANE et al., 2021). Este avanço tecnológico é evidenciado pela evolução dos sistemas de orientação dos AGVs ao longo do tempo, conforme mostrado na Figura 39. As fases incluem: (a) navegação mecânica, (b) navegação óptica, (c) navegação indutiva, (d) referência cartesiana, (e) navegação inercial e, (f) navegação laser (FRAGAPANE et al., 2021).

Figura 39 – Sistemas de orientação utilizados por AGVs, vistos a partir de uma vista superior



Fonte: Adaptado de Fragapane et al. (2021)

De fato, segundo Groover (2008), existem diferentes categorias de sistemas de orientação que se dividem em duas categorias principais: ativos e passivos. Entre os ativos, destacam-se a navegação indutiva (Figura 39c), inercial (Figura 39e) e a laser (Figura 39f). Estes sistemas

dependem de tecnologias que ativamente guiam e ajustam a trajetória do veículo, como por exemplo, cabos enterrados no chão que emitem campos magnéticos.

Por outro lado, os sistemas passivos, exemplificados pelas navegações mecânica (Figura 39a) e óptica (Figura 39b), operam com base em guias ou marcas físicas presentes no ambiente, as quais são passivamente seguidas pelos AGVs sem a necessidade de ajustes ativos na trajetória. Tais sistemas utilizam, por exemplo, fitas magnéticas ou faixas pintadas no chão. Sacchetin (2005) caracteriza os sistemas passivos como “reativos”, destacando sua capacidade de identificar e reagir a diferenças imediatas entre a posição atual e a posição alvo. Embora eficientes, esses sistemas enfrentam a limitação de não possuírem informações antecipadas sobre os percursos futuros, necessitando assim da integração com sistemas externos para adquirir e centralizar tais dados. Para isso, eles dependem de uma unidade central, conhecida como gerenciador de frotas (ou *fleet manager*, em inglês), que está conectada a diversos serviços. Este gerenciador de frotas é encarregado de tomar decisões de controle, tais como determinar as rotas a serem seguidas e coordenar o despacho de novos AGVs, com base nas informações recebidas (FRAGAPANE et al., 2021).

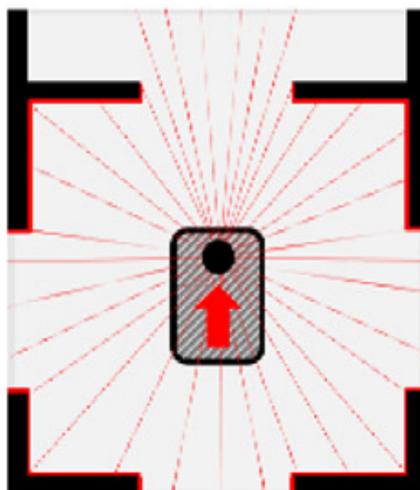
Em termos de funcionalidade, os AGVs são programados para seguir regras e caminhos específicos, com autonomia limitada nas tomadas de decisões. Portanto, estes veículos são frequentemente utilizados para tarefas repetitivas e previsíveis, transportando cargas horizontalmente de um ponto para outro, seja em armazéns, ambientes de manufatura ou portos (LE-ANH; KOSTER, 2006).

Adicionalmente, Fragapane et al. (2021) destacam que os AGVs apresentam mais limitações acerca de sua flexibilidade. A dependência de rotas fixas e a necessidade de infraestrutura dedicada para operação dos veículos, como pintura do piso e instalações de fitas guia, podem representar desafios, especialmente em ambientes que exigem alta adaptabilidade. Isso significa que qualquer mudança no *layout* do percurso exige alterações físicas no ambiente, o que, por sua vez, pode resultar na interrupção das operações e acarretar em custos adicionais.

Diferente dos sistemas descritos anteriormente, os sistemas baseados em visão (Figura 40) utilizados pelos AMRs não requerem trajetos fisicamente definidos. Esses sistemas fazem o uso de diversos sensores, computadores de bordo, Inteligência Artificial (IA) e tecnologia *Simultaneous Localization and Mapping* (SLAM), permitindo que o robô reconheça o ambiente sem a obrigação de ter rotas previamente definidas ou alterações na infraestrutura para planejar e implementar pontos de referência (FRAGAPANE et al., 2021).

Para alcançar a navegação autônoma, interação com o ambiente e capacidade de tomar decisões, as principais tecnologias utilizadas em AMRs incluem tecnologias de sensoriamento, de modo a incorporar uma variedade de sensores de baixo custo com uma boa eficiência, como *scanners Light Detection and Ranging* (LiDAR) para geração precisa de nuvens de pontos que refletem o ambiente ao seu redor, câmeras 3D para reconhecimento visual e amplo de obstáculos, acelerômetros e giroscópios para detecção de movimento e identificação da orientação do robô,

Figura 40 – Sistema de orientação baseado em visão utilizado pelos AMRs



Fonte: Adaptado de Fragapane et al. (2021)

e codificadores de roda para calcular distâncias percorridas e suas posições (SILVA; ROCHE; KONDOZ, 2018).

Além disso, os AMRs utilizam a tecnologia SLAM para entender o seu atual ambiente e criar mapas de navegação em tempo real sem depender de pontos de referência predefinidos (ver, Bloss (2008)). Isso permite que eles planejem suas próprias rotas e se adaptem a mudanças nas condições do ambiente, como por exemplo, alterar a atual rota devido a caminhos bloqueados ou evitar obstáculos.

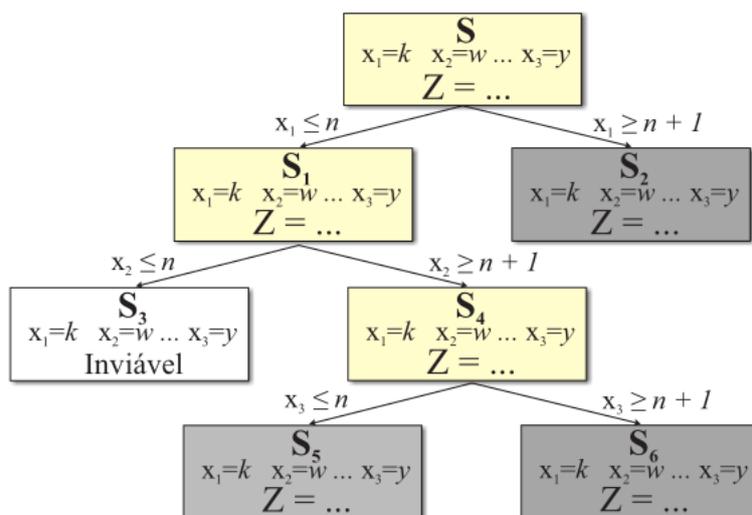
Uma variedade de técnicas de fusão entre IA, Aprendizado de Máquina (ML), lógica *fuzzy*, redes neurais, algoritmos *neuro-fuzzy* e genéticos, podem ser combinadas para permitir que os AMRs se desloquem de maneira eficiente e segura desde o início de sua jornada até o destino final, sem colidirem com obstáculos ou serem interrompidos (DIAS et al., 2018).

APÊNDICE D – FUNCIONAMENTO DO MÉTODO *BRANCH-AND-BOUND*

Para explicar os métodos *branching* e *bounding* de forma mais prática, considere que P simboliza um problema de minimização a ser resolvido, enquanto S_0 é um subconjunto de S , no qual contém todas as soluções viáveis de P . Inicialmente, o nó raiz é resolvido para obter uma solução inicial, para isso, se estabelece um limite inferior para uma versão simplificada do problema, denominada forma relaxada, na qual as restrições de integralidade são desconsideradas, e o problema é resolvido utilizando alguma técnica de programação linear contínua, como o método *Simplex* ou o *Dual Simplex*. Se a solução inicial resultante for inteira, ou seja, se todas as variáveis de decisão encontradas por meio do método aplicado forem inteiras, então o problema é considerado como resolvido. Porém, se isso não acontecer, é iniciado o processo de *branching* para dividir o problema em instâncias menores (BRUCKER, 2007).

Quando o processo de *branching* é executado, o conjunto S é dividido em problemas menores, representados por $S_i (i = 1, 2, \dots, n)$. Cada um desses subconjuntos S_i é representado em uma estrutura de árvore de recursão, chamada de *branch tree*. Nessa estrutura, S é o nó-raiz, e cada um dos subproblemas S_i são nós originados de S . Da mesma forma, cada um dos subproblemas S_i tem a possibilidade de gerar novas sub-árvores de problemas. Ao adicionar um novo nó, as restrições sobre a variável de decisão específica é atualizada de acordo com as restrições de ramificação introduzidas no nó pai, de modo a melhorar o espaço de busca (BRUCKER, 2007). A Figura 41 ilustra um exemplo de árvore de *branch-and-bound*, destacando os processos de poda e ramificação.

Figura 41 – Exemplo de árvore de recursão *Branch-and-Bound*



Fonte: Silva (2018)

Conforme o processo de *branching* avança, ocorre o processo de *bounding* (definição de limites), que determina a continuidade ou interrupção do método. Se uma restrição adicionada a um subproblema o tornar inviável, o nó é podado e não há novas ramificações. Quando o limite

inferior de um subproblema é maior ou igual ao limite superior global dado por UB , o subproblema não é ramificado, pois não pode melhorar a solução. Caso restem variáveis de decisão contínuas, o *branching* continua, criando novos subproblemas com novas restrições. Quando um nó produz uma solução inteira, ele não é ramificado novamente, e sua solução é considerada a melhor solução inteira atual, atualizando BU . O processo é concluído quando todos os subproblemas foram resolvidos, ou seja, quando não é possível gerar mais nós na árvore. Nesse ponto, BU representa a solução ótima para o problema (BRUCKER, 2007). Uma versão simplificada do algoritmo *branch-and-bound* pode ser vista no Algoritmo 1 (BRUCKER, 2007).

Algorithm 1 *Branch-and-Bound*

```

1:  $Q \leftarrow S$ 
2:  $UB \leftarrow$  Valor da melhor solução atual (limite superior)
3:  $bestCurrent \leftarrow$  Melhor solução atual
4: while  $Q \neq \emptyset$  do
5:   Seleciona um nó de ramificação  $k$  de  $Q$ 
6:   Remove  $k$  de  $Q$ 
7:   Gere filhos  $C_i$  (para  $i = 1, 2, \dots, n_k$ )
8:   for  $i \leftarrow 1$  até  $n_k$  do
9:     Calcule os limites inferiores correspondentes  $LB_i$ 
10:    if  $LB_i < UB$  then
11:      if  $C_i$  é uma solução inteira then
12:        if  $LB_i$  é uma solução melhor que  $UB$  then
13:           $UB \leftarrow LB_i$ 
14:           $bestCurrent \leftarrow$  Recebe solução correspondente de  $C_i$ 
15:        end if
16:      else
17:        Adicione  $C_i$  para  $Q$ 
18:      end if
19:    end if
20:  end for
21: end while

```

No Algoritmo 1, inicialmente define-se Q como o conjunto de subproblemas não explorados (soluções candidatas) e um limite superior (UB) que representa o valor da melhor solução encontrada até o momento. O procedimento segue um ciclo iterativo enquanto existirem elementos em Q . A cada iteração, um nó de ramificação k é selecionado e removido de Q , e então são gerados filhos C_i , que representam subdivisões (ramificações) do problema original. Para cada filho C_i , o limite inferior LB_i é calculado, servindo como uma estimativa para a qualidade da solução derivada deste subproblema. Se C_i corresponde a uma solução inteira e seu LB_i for maior que o valor atual de BU , então BU e a melhor solução atual, $bestCurrent$, são atualizados. Caso C_i não seja uma solução inteira, mas apresente um LB_i promissor (inferior a BU), este subproblema é adicionado novamente ao conjunto Q para que uma nova análise seja feita. Esse processo é repetido até que Q esteja vazio, resultando na obtenção da melhor solução possível para o problema.

Os métodos de implementação para enumeração e busca na árvore são flexíveis, permitindo modificações e diferentes estratégias de implementação. Duas regras geralmente usadas são: *Depth First Search* (DFS), conhecida como, *Last In First Out* (LIFO) que prioriza a exploração em profundidade da árvore, e *Breadth First Search* (BFS), que prioriza a exploração em largura (BRUCKER, 2007).

APÊNDICE E – FUNCIONAMENTO DOS PLANOS DE CORTE E DO MÉTODO *BRANCH-AND-CUT*

O método *Branch-and-Cut* utiliza planos de corte como parte fundamental de sua estratégia. Por sua vez, os planos de cortes são estratégias utilizadas para melhorar a eficiência do processo de busca pela solução ótima, sua aplicação envolve algumas etapas. Inicialmente, a relaxação do modelo de PL é resolvida, permitindo que as variáveis de decisão assumam valores contínuos. Em seguida, a solução relaxada é examinada para encontrar desigualdades que são violadas, ou seja, soluções que são fracionárias. Quando uma solução fracionária é encontrada, um plano de corte é gerado e incorporado à relaxação. Este plano é uma nova restrição que “corta” a solução fracionária do espaço de solução, mas não exclui nenhuma solução inteira viável tornando-a mais restritiva. O processo é iterativo, com a relaxação e os planos de corte adicionados sendo resolvidos repetidamente até que uma solução inteira seja alcançada ou até que se prove que a relaxação PL é inviável. Ao adicionar o plano de corte, o espaço de solução é reduzido, permitindo que o algoritmo se concentre nas áreas que ainda podem conter a solução ótima (HOFFMAN; RALPHS, 2013). Uma versão simplificada do algoritmo de planos de cortes pode ser vista no Algoritmo 2.

Algorithm 2 Algoritmo de Planos de Cortes

```

1:  $P_0 \leftarrow$  Formulação inicial do problema (relaxada)
2:  $t \leftarrow 0$ 
3: repeat
4:    $z_t \leftarrow$  Solução da formulação  $P_t$ 
5:   if  $z_t$  não é uma solução inteira then
6:      $d \leftarrow$  desigualdade para eliminar  $z_t$  do problema
7:      $P_{t+1} \leftarrow P_t$  mais desigualdade  $d$ 
8:      $t \leftarrow t + 1$ 
9:   end if
10: until  $z_t$  seja uma solução inteira

```

No Algoritmo 2, inicia-se com a formulação relaxada P_0 eliminando temporariamente as restrições de integralidade para simplificar a busca pela solução ótima. Com t inicializado em zero, o algoritmo avança iterativamente, resolvendo P_t para obter z_t . Caso z_t não satisfaça os critérios de integralidade, um plano de corte (uma desigualdade) d é gerado para excluir z_t sem eliminar soluções inteiras viáveis, refinando assim o espaço de busca. Esse plano de corte é então adicionado à formulação atual, criando P_{t+1} , uma versão mais restritiva do problema. Posteriormente, incrementa-se t , e o processo continua até que z_t seja uma solução inteira, retornando a descoberta da solução ótima.

Uma variedade de tipos de planos de corte, incluindo cortes de Gomory (1958), cortes de levantamento e projeção (BALAS; NG, 1989), e cortes de arredondamento misto (NEMHAUSER; WOLSEY, 1988), podem ser aplicados de acordo com a estrutura do problema e a abor-

dagem de solução específica adotada.

Enquanto o Algoritmo 2 ilustra o processo de adicionar planos de corte a um problema relaxado, o método *Branch-and-Cut* expande essa abordagem ao incorporar os planos de corte diretamente na árvore de decisão do *Branch-and-Bound*. Uma versão simplificada do algoritmo *Branch-and-Cut* pode ser vista no Algoritmo 3.

Algorithm 3 Algoritmo *branch-and-cut*

```

1:  $Q \leftarrow S$ 
2:  $UB \leftarrow$  Valor da melhor solução atual (upper bound)
3:  $bestCurrent \leftarrow$  Melhor solução atual
4: while  $Q \neq \emptyset$  do
5:   Seleciona um nó de ramificação  $k$  de  $L$ 
6:   Remove  $k$  de  $L$ 
7:   Aplica plano de corte sobre  $k$ 
8:   Gerar os filhos  $C_i$  (para  $i = 1, 2, \dots, n_k$ )
9:   for  $i \leftarrow 1$  até  $n_k$  do
10:    Calcular os limites inferiores correspondentes  $LB_i$ 
11:    if  $LB_i < UB$  then
12:      if  $C_i$  é uma solução inteira then
13:        if  $LB_i$  é uma solução melhor que  $UB$  then
14:           $UB \leftarrow LB_i$ 
15:           $bestCurrent \leftarrow$  Solução correspondente de  $C_i$ 
16:        end if
17:      else
18:        if  $LB_i$  é uma solução melhor que  $UB$  then
19:          Adiciona  $C_i$  para  $Q$ 
20:        end if
21:      end if
22:    end if
23:  end for
24: end while

```

O Algoritmo 3 funciona através da exploração sistemática do espaço de busca e utiliza estratégias de ramificação, cortes e limites para encontrar a melhor solução. Para isso, inicialmente, ele mantém um conjunto de soluções candidatas representado por Q e um limite superior (UB) que representa a melhor solução encontrada até o momento. O algoritmo então seleciona um nó de ramificação k de Q , remove k de Q , e aplica planos de corte sobre k para reduzir o espaço de busca. Em seguida, gera filhos C_i a partir desse nó e calcula limites inferiores correspondentes LB_i para cada filho. Se LB_i for menor que UB , ele verifica se C_i é uma solução inteira. Se for inteira e se LB_i for melhor que UB , UB e a melhor solução atual $bestCurrent$ são atualizados. Se C_i não for uma solução inteira, mas seu LB_i for promissor, ele é adicionado de volta a Q . Esse processo é repetido até que Q esteja vazio, resultando na obtenção da melhor solução possível para o problema.

APÊNDICE F – PSEUDO-CÓDIGO *GUIDED LOCAL SEARCH*

Algorithm 4 *Guided Local Search*

```

1:  $k \leftarrow 0$ 
2:  $s_1 \leftarrow \text{Solucao\_Inicial}(); s' \leftarrow s_1$ 
3: for  $i = 1$  to  $M$  do
4:    $p_i \leftarrow 0$ 
5: end for
6: while critério de parada não satisfeito do
7:    $s_{k+1} \leftarrow \text{BuscaLocal}(s_k, h)$ 
8:   if  $g(s_{k+1}) < g(s')$  then
9:      $s' \leftarrow s_{k+1}$ 
10:  end if
11:  for  $i = 1$  to  $M$  do
12:     $util_i \leftarrow \frac{I_i(s_{k+1})c_i}{1+p_i}$ 
13:  end for
14:  for  $i = 1$  to  $M$  do
15:     $i \leftarrow \text{argmax}_i(util_i)$ 
16:     $p_i \leftarrow p_i + 1$ 
17:  end for
18:  if  $k = \text{Zera\_Penalidades}$  then
19:    for  $i = 1$  to  $M$  do
20:       $p_i \leftarrow 0$ 
21:    end for
22:     $k \leftarrow 0$ 
23:  end if
24:   $k \leftarrow k + 1$ 
25: end while
26: return  $s'$ 

```

No Algoritmo 4 (VOUDOURIS, 1997), o processo começa com a inicialização, definindo o contador de iterações k como 0, gerando uma solução inicial s_1 e estabelecendo ela como a melhor solução s' até o momento. Em seguida, todas as penalidades p_i para cada característica/atributo são inicializadas como zero. Posteriormente, o processo entra em um *loop* de busca local, partindo da solução atual s_k , que utiliza a função h como referência para gerar uma nova solução s_{k+1} . Se $g(s_{k+1})$, que é o valor da função objetivo da nova solução, for menor que $g(s')$, então s_{k+1} se torna a nova melhor solução s' . Para cada característica, o algoritmo calcula a *utilidade*, considerando o custo da característica, as penalidades atuais e o impacto na solução, identificando quais características estão contribuindo mais para o custo da solução. A característica com o valor *utilidade* mais alto é penalizada, incrementando sua penalidade p_i em 1, de modo a evitar a seleção de características que aumentam significativamente o custo da solução. Caso k alcance o valor de *Zera_Penalidades*, todas as penalidades voltam para zero, o que previne que o algoritmo fique preso em mínimos locais por penalizar características específicas muitas vezes. O contador k é incrementado, e o processo se repete até que um critério de

parada seja satisfeito, retornando a melhor solução s' encontrada.

O resultado do processo descrito pode ser visto graficamente através da Figura 15, que representa um problema de minimização, onde a solução inicial é representada pelo ponto $x = a$ e posteriormente o processo executa uma busca local e atinge um ótimo local, marcado pelo ponto $x = b$. Ao alcançar esse ótimo local, a GLS penaliza as características/atributos da solução encontrada. Essas penalidades alteram a função objetivo, essa transformação é visualmente representada pelas linhas vermelhas. A aplicação de novas penalidades faz com que o ponto $x = b$ deixe de ser considerado um ótimo local (evidenciado pela linha vermelha superior), incentivando a busca a procurar por uma nova localização ótima. Esse movimento da busca local é simbolizado pela linha cinza, que destaca a transição do ponto $x = b$, agora deixando de ser um ótimo local, para atingir o ponto $x = c$, identificado como o novo ótimo local.

APÊNDICE G – PSEUDO-CÓDIGO DO ALGORITMO DE DIJKSTRA

Algorithm 5 Algoritmo de Dijkstra

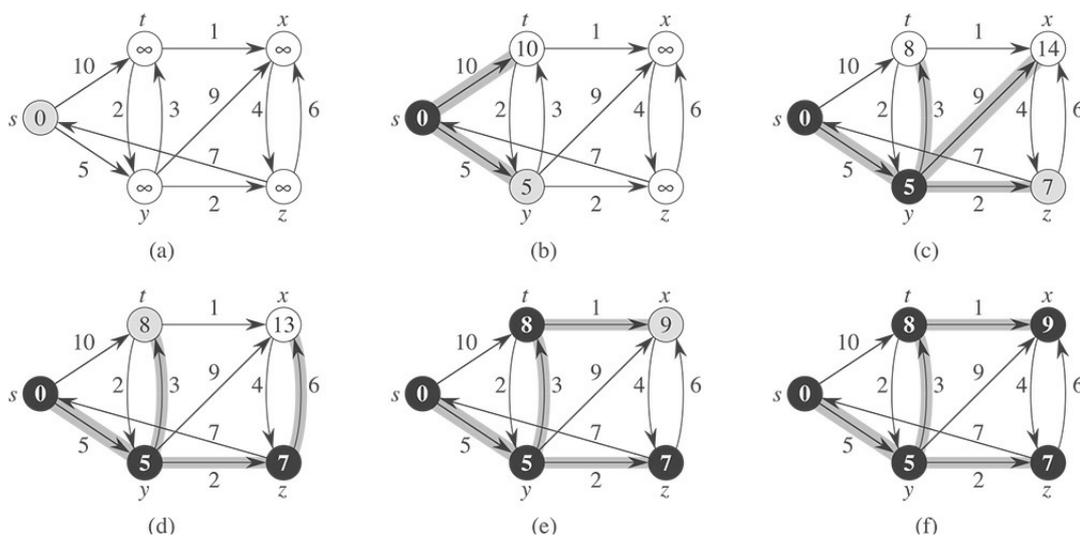
```

1: Entrada: Grafo  $G$ , função de peso  $w$ , vértice de origem  $s$ 
2: Saída: Distância mínima  $d[]$ , predecessores  $prev[]$ 
3:
4: procedure DIJKSTRA( $G, w, s$ )
5:   for cada vértice  $v \in V(G)$  do
6:      $d[v] \leftarrow \infty$ 
7:      $prev[v] \leftarrow$  indefinido
8:   end for
9:    $d[s] \leftarrow 0$ 
10:   $Q \leftarrow V(G)$ 
11:   $S \leftarrow \emptyset$ 
12:  while  $Q \neq \emptyset$  do
13:     $u \leftarrow$  extract-min( $Q$ )
14:     $S \leftarrow S \cup \{u\}$ 
15:    for cada vértice  $v \in G.Adj[u]$  do
16:       $alt \leftarrow d[u] + w(u, v)$ 
17:      if  $alt < d[v]$  then
18:         $d[v] \leftarrow alt$ 
19:         $prev[v] \leftarrow u$ 
20:      end if
21:    end for
22:  end while
23:  return  $d[], prev[]$ 
24: end procedure

```

O Algoritmo 5 recebe como entrada um grafo G , uma função de peso w , e um vértice de origem s . Inicialmente, o algoritmo define a distância de todos os vértices como infinita (∞) e seus predecessores como indefinidos, exceto o vértice de origem s , que recebe a distância de zero, uma vez que é a distância até ele mesmo. A variável Q é uma fila de prioridades que contém o conjunto de todos os vértices do grafo, enquanto o conjunto S representa os vértices já visitados, sendo inicializado como vazio. O algoritmo iterativamente seleciona o vértice u de Q com a menor distância atual (utilizando a função *extract-min* que remove um elemento com menor chave da fila), retirando-o de Q e posteriormente adicionando-o em S . Em seguida, para cada vértice v adjacente a u , o algoritmo calcula a distância alternativa (*alt*) para alcançar v saindo de u , somando a distância atual de u ($d[u]$) e o peso da aresta que conecta u a v ($w(u, v)$). Se *alt* for menor que a distância atual de v , $d[v]$ é atualizado para *alt*, e u é definido como o predecessor de v . Esse processo continua até que todos os vértices tenham sido visitados, retornando como saída a distância mínima $d[]$ de s para todos os outros vértices e um *array* de predecessores $prev[]$ que representam os caminhos mínimos a partir do vértice de origem, possibilitando reconstruir o caminho mais curto de s a cada vértice.

Figura 42 – Representação da execução do algoritmo de Dijkstra



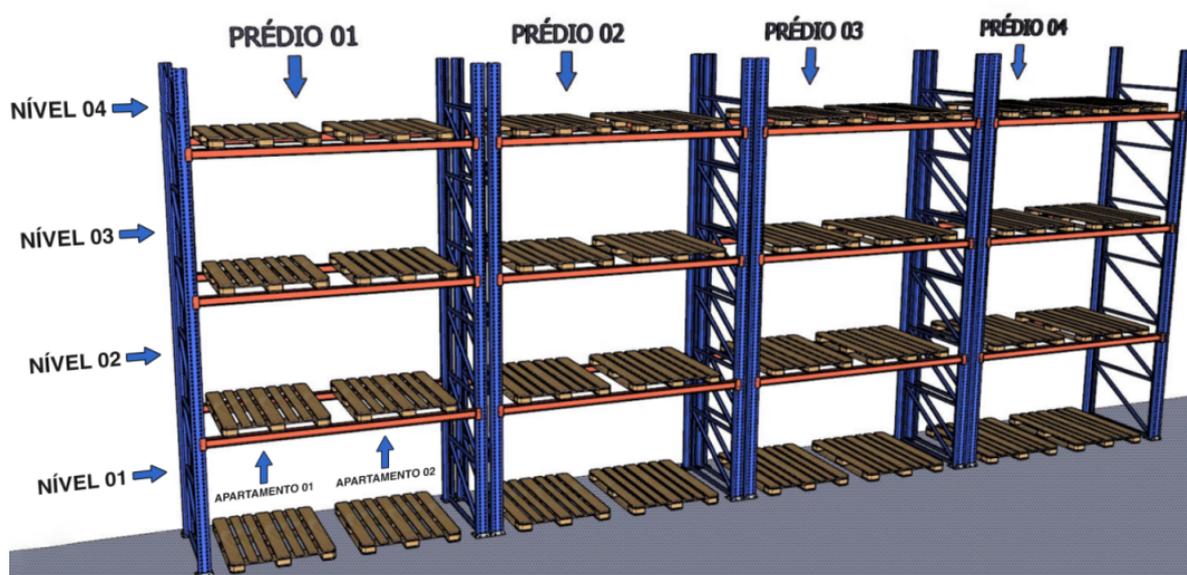
Fonte: Cormen et al. (2009)

O processo para encontrar o caminho mínimo realizado pelo algoritmo Dijkstra pode ser visto na Figura 42. Na figura, as estimativas de distância mínima de s até todos os outros vértices são representadas dentro dos próprios vértices, enquanto as arestas que levam ao predecessor de cada vértice no caminho mínimo estão destacadas por um sombreamento. O conjunto S é representado pelos vértices pretos, enquanto o conjunto Q , pelos vértices brancos. Antes da primeira iteração representada pela Figura (a), apenas o vértice fonte s tem uma distância mínima definida (zero), e é marcado para ser o próximo vértice u a ser explorado. Durante cada iteração do laço *while* (Figuras b a f), o algoritmo seleciona o vértice u de Q com a menor distância até o momento, o qual é então movido para o conjunto S , indicando que a menor distância de s a u foi encontrada.

APÊNDICE H – CONCEITOS SOBRE A ESTRUTURA DE ARMAZENAMENTO

Para padronizar algumas informações e facilitar a compreensão dos conceitos utilizados nesta dissertação, serão explicados a seguir alguns termos-chave. Esses termos são fundamentais para entender a metodologia aplicada e os resultados obtidos ao longo deste trabalho. Portanto, ao se tratar das estruturas de armazenagem na qual os produtos são armazenados, a Figura 43 demonstra as principais nomenclaturas utilizadas:

Figura 43 – Estrutura porta *pallet* exemplificando nomenclaturas utilizados



Fonte: O Autor (2024)

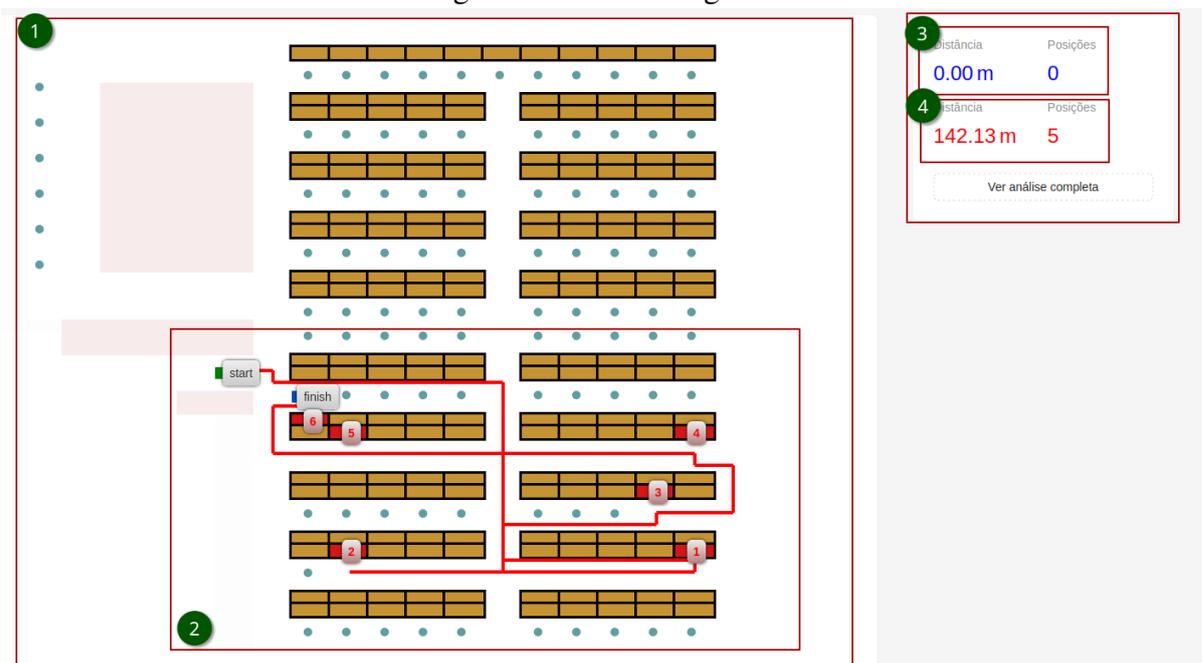
- **Prédio:** Refere-se à estrutura de uma seção de porta *pallet* ou prateleira que vai de uma longarina vertical até a outra;
- **Nível:** Cada “andar” ou “prateleira” dentro de um prédio. Os níveis são definidos pela posição das longarinas e são ajustáveis em altura para acomodar diferentes tamanhos de *pallets*;
- **Apartamento:** Refere-se a divisão de diferentes endereços dentro de cada nível de um prédio. Os apartamentos são as posições onde os *pallets* são armazenados. Cada apartamento é um endereço único.

As nomenclaturas apresentadas acima não se limitam exclusivamente às estruturas de porta *pallets*. Elas podem ser aplicadas a outros tipos de estruturas de armazenagem, como prateleiras, estantes, entre outras, onde o conceito de divisão em níveis, apartamentos e prédios se faz necessário para uma organização lógica e eficiente dos produtos armazenados.

APÊNDICE I – INTERFACE GRÁFICA

Para facilitar a construção e edição da representação virtual do armazém, assim como a realização dos experimentos e a análise dos resultados, foi desenvolvida uma interface gráfica, disponibilizada como uma aplicação web. Essa interface utiliza o mapeamento virtual para representar o *layout* do armazém da empresa em um formato de mapa bidimensional. Para mais detalhes, consultar o Apêndice K.

Figura 44 – Interface gráfica



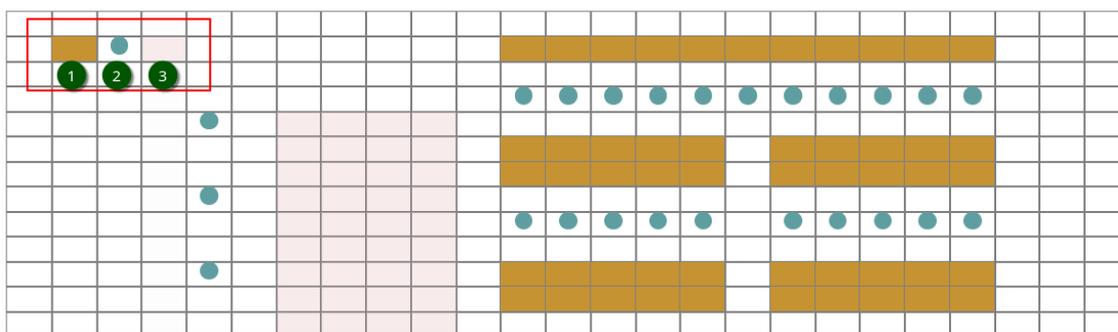
Fonte: O Autor (2024)

A Figura 44 apresenta a interface gráfica desenvolvida, na qual:

- O item 1 refere-se ao *layout* da área de *picking* do armazém, onde as estruturas de porta *pallets* são representadas em laranja, e os pontos de coleta são indicados por círculos verdes;
- O item 2 exibe uma possível rota de separação, composta por uma sequência de endereços numerados de acordo com a ordem a ser seguida, com destaque para os pontos inicial (*start*) e final (*finish*) da rota. Rotas em vermelho indicam que ela foi planejada pelo WMS. Rotas em azul indicam que foram rotas propostas pela ferramenta de otimização;
- O item 3 fornece valores da rota otimizada (azul), que incluem a distância percorrida e o número de pontos de coleta visitados;
- O item 4 apresenta valores da rota planejada (vermelho), igualmente compreendendo a distância percorrida e a quantidade de pontos de coleta visitados.

Para auxiliar na construção ou modificação do *layout*, a ferramenta dispõe de um modo de edição que permite ajustar e desenvolver a representação do *layout* do armazém de acordo com as necessidades específicas. Quando este modo está ativado, a malha do *grid* torna-se visível, facilitando a adição ou remoção de elementos, como pontos de coleta, prédios ou obstáculos, nas posições desejadas.

Figura 45 – Modo de construção/edição da ferramenta

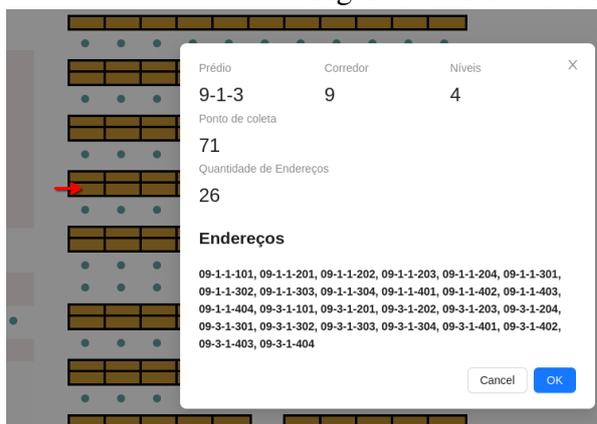


Fonte: O Autor (2024)

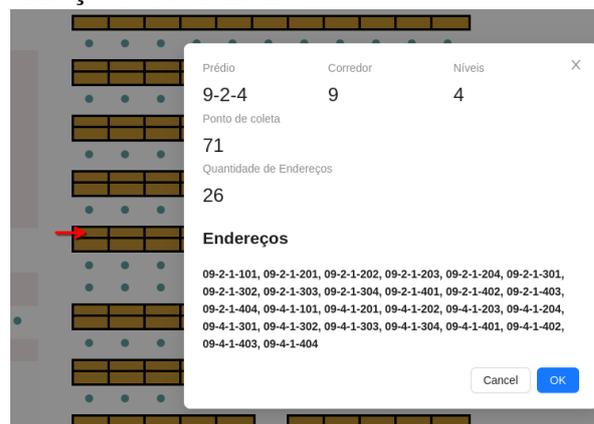
Na Figura 45, observa-se o modo de construção/edição habilitado. O item 1 refere-se a um elemento do tipo prédio, o item 2 a um elemento do tipo ponto de coleta, e o item 3 corresponde a um elemento do tipo obstáculo. Cada um desses elementos pode ser selecionado e posicionado no armazém conforme necessário. Além do modo de construção, a ferramenta também oferece um modo de informações, que disponibiliza detalhes sobre os elementos mapeados no armazém. Este modo permite visualizar informações relevantes sobre pontos de coleta (Figura 46c), prédios (Figuras 46a e 46b) e possibilita também a análise das distâncias entre pontos de coleta de interesse (Figura 46d).

Por fim, a ferramenta disponibiliza funcionalidades para a realização de análises comparativas entre rotas submetidas à otimização e as rotas sugeridas. Essas análises incluem tanto comparações individuais de uma única rota de coleta, destacando cada endereço (Figura 47a), quanto uma análise geral (Figura 47b) e individual (Figura 47c) de períodos específicos, selecionados conforme o interesse.

Figura 46 – Modo de informação da ferramenta



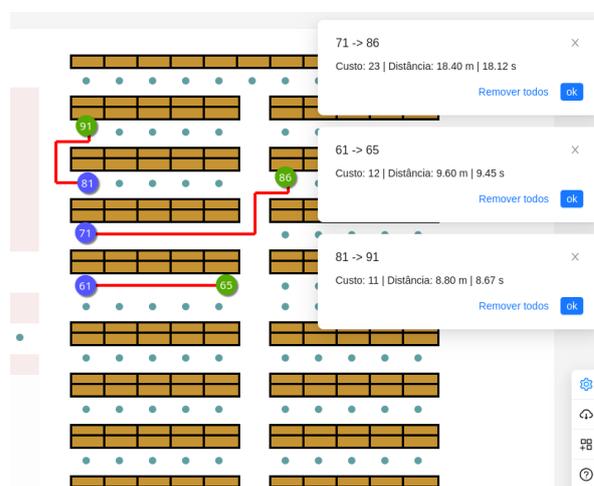
(a) Informações sobre o prédio 9-1-3



(b) Informações sobre o prédio 9-2-4



(c) Informações sobre o ponto de coleta 71



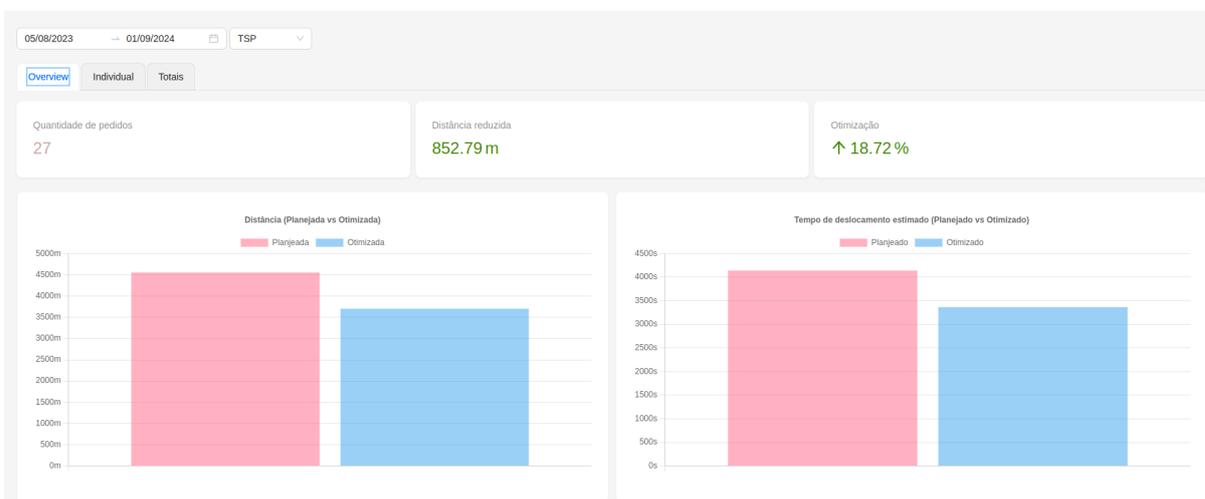
(d) Distâncias entre pontos de coleta

Fonte: O Autor (2024)

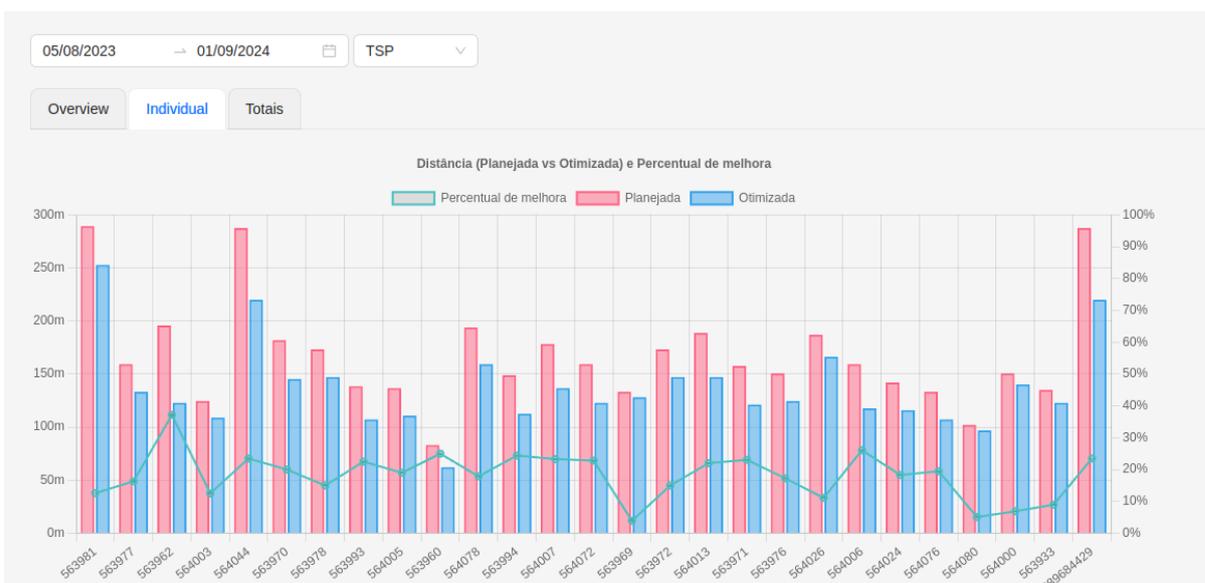
Figura 47 – Análises comparativas fornecidas pela ferramenta



(a) Comparação individual a cada endereço em uma rota de picking



(b) Comparação geral de um período específico



(c) Comparação individual de um período específico

Fonte: O Autor (2024)

APÊNDICE J – MÉTODO DE CONSTRUÇÃO E CÁLCULO DA MATRIZ DE DISTÂNCIAS

Com base nos pontos de coleta mapeados, foi criada uma matriz de distâncias, onde cada posição contém a medida da menor distância entre um par específico de pontos de coleta. Para construir essa matriz, utilizou-se a representação bidimensional virtual do armazém. As distâncias entre cada par de pontos foram calculadas utilizando o algoritmo de Dijkstra, que determina a menor distância entre um ponto de origem e todos os outros pontos, considerando obstáculos e restrições do armazém. O algoritmo utiliza as coordenadas (x, y) dos pontos de origem e destino como dados de entrada. Inicialmente, atribui-se ao nó de origem uma distância zero, enquanto atribui a todas as outras células uma distância considerada infinita, indicando que ainda não foram visitadas. Em seguida, o algoritmo explora os nós adjacentes (células vizinhas) ao nó atual, movendo-se apenas nas direções horizontal e vertical. Para cada célula adjacente que não seja um obstáculo ou um prédio, a distância acumulada até este ponto é atualizada com base no custo do movimento associado, se o caminho até ela representar uma distância menor do que qualquer outra previamente calculada. As células ainda não visitadas são armazenadas em uma fila de prioridades, onde a prioridade é inversamente proporcional à distância acumulada, garantindo que o nó com a menor distância seja escolhido para a próxima exploração, o que indica a exploração do caminho potencialmente mais curto. Esse procedimento é repetido até que o ponto de coleta de destino seja alcançado. Por fim, a rota mais curta entre os dois pontos de coleta é determinada por meio de um processo de *backtracking*, que retrocede do ponto de destino até o ponto de origem, seguindo as referências de precedência estabelecidas a cada iteração do algoritmo.

Para preencher a matriz de distâncias, o processo anteriormente descrito deve ser repetido para cada par de pontos de coleta da matriz 2D do armazém. A matriz de distâncias resultante deve ser quadrada e sua ordem corresponde ao número total de pontos de coleta, N . No entanto, se as distâncias entre os pontos de coleta forem simétricas, isso é, a distância do ponto 1 para o ponto 2 é igual a distância do ponto 2 até o ponto 1, então é suficiente calcular apenas a metade superior da matriz. Para isso, emprega-se duas iterações aninhadas e para cada ponto de coleta i , calcula-se as distâncias somente para os pontos de coleta j , onde $j > i$. Este método resulta em $\frac{N \times (N-1)}{2}$ cálculos de distâncias, cada um representado um par único de pontos de coleta, garantindo a cobertura completa das distâncias necessárias sem redundâncias e em um tempo computacional menor. A Figura 48 apresenta uma amostra dos resultados obtidos ao aplicar o método nos dados reais da empresa.

Por fim, a matriz de distâncias resultante armazena o custo associado a cada deslocamento, diferenciado conforme a direção do movimento (horizontal ou vertical). Cada transição de movimento entre células adjacentes deve ter um custo específico associado. Posteriormente, para converter esses custos em uma estimativa das distâncias reais percorridas, os valores devem ser ajustados multiplicando-se por uma constante de proporcionalidade. Esta constante deve ser

Figura 48 – Matriz de distâncias

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	
1	0	1	2	3	4	6	7	8	9	10	7	8	9	10	11	11	12	13	14	15	12	13	14	15	16	16	17	18	19	20	17	18	
2		0	1	2	3	5	6	7	8	9	8	9	10	11	10	10	11	12	13	14	13	14	15	16	15	15	16	17	18	19	18	18	19
3			0	1	2	4	5	6	7	8	9	10	11	10	9	9	10	11	12	13	14	15	16	15	14	14	15	16	17	18	19	20	
4				0	1	3	4	5	6	7	10	11	10	9	8	8	9	10	11	12	15	16	15	14	13	13	14	15	16	17	20	21	
5					0	2	3	4	5	6	11	10	9	8	7	7	8	9	10	11	16	15	14	13	12	12	13	14	15	16	21	20	
6						0	1	2	3	4	11	10	9	8	7	7	8	9	10	11	16	15	14	13	12	12	13	14	15	16	21	20	
7							0	1	2	3	12	11	10	9	8	8	9	10	11	10	17	16	15	14	13	13	14	15	16	15	22	21	
8								0	1	2	13	12	11	10	9	9	10	11	10	9	18	17	16	15	14	14	15	16	15	14	23	22	
9									0	1	14	13	12	11	10	10	11	10	9	8	19	18	17	16	15	15	16	15	14	13	24	23	
10										0	15	14	13	12	11	11	10	9	8	7	20	19	18	17	16	16	15	14	13	12	25	24	
11											0	1	2	3	4	6	7	8	9	10	7	8	9	10	11	11	12	13	14	15	12	13	
12												0	1	2	3	5	6	7	8	9	8	9	10	11	10	10	11	12	13	14	13	14	
13													0	1	2	4	5	6	7	8	9	10	11	10	9	9	10	11	12	13	14	15	
14														0	1	3	4	5	6	7	10	11	10	9	8	8	9	10	11	12	15	16	
15															0	2	3	4	5	6	11	10	9	8	7	7	8	9	10	11	16	15	
16																0	1	2	3	4	11	10	9	8	7	7	8	9	10	11	16	15	
17																	0	1	2	3	12	11	10	9	8	8	9	10	11	10	17	16	
18																		0	1	2	13	12	11	10	9	9	10	11	10	9	18	17	
19																			0	1	14	13	12	11	10	10	11	10	9	8	19	18	
20																				0	15	14	13	12	11	11	10	9	8	7	20	19	
21																					0	1	2	3	4	6	7	8	9	10	7	8	
22																						0	1	2	3	5	6	7	8	9	8	9	
23																							0	1	2	4	5	6	7	8	9	10	
24																								0	1	3	4	5	6	7	10	11	
25																									0	2	3	4	5	6	11	10	
26																										0	1	2	3	4	11	10	
27																											0	1	2	3	12	11	
28																												0	1	2	13	12	
29																													0	1	14	13	
30																														0	15	14	
31																															0	1	
32																																0	

Fonte: O Autor (2024)

derivada da escala do mapa 2D, estabelecida durante a fase de mapeamento e que corresponde às dimensões reais do armazém.

APÊNDICE K – MODELAGEM VIRTUAL DO ARMAZÉM

A modelagem virtual do armazém foi realizada a partir do mapeamento físico detalhado de suas estruturas. Para isso, inicialmente, identificaram-se os endereços de armazenamento existentes, que foram registrados em uma planilha contendo informações que indicam o número do prédio, apartamento e nível (Apêndice H). Paralelamente, foram mapeados os pontos de coleta, que representam as posições estratégicas onde os robôs realizam as paradas para atender os endereços de coleta. Para uma descrição detalhada da motivação e do processo de concepção desses pontos de coleta, consultar o Apêndice L. A Figura 49 apresenta a estrutura da planilha de mapeamento utilizada, destacando os principais dados coletados.

Figura 49 – Estrutura de mapeamento dos endereços de armazenagem

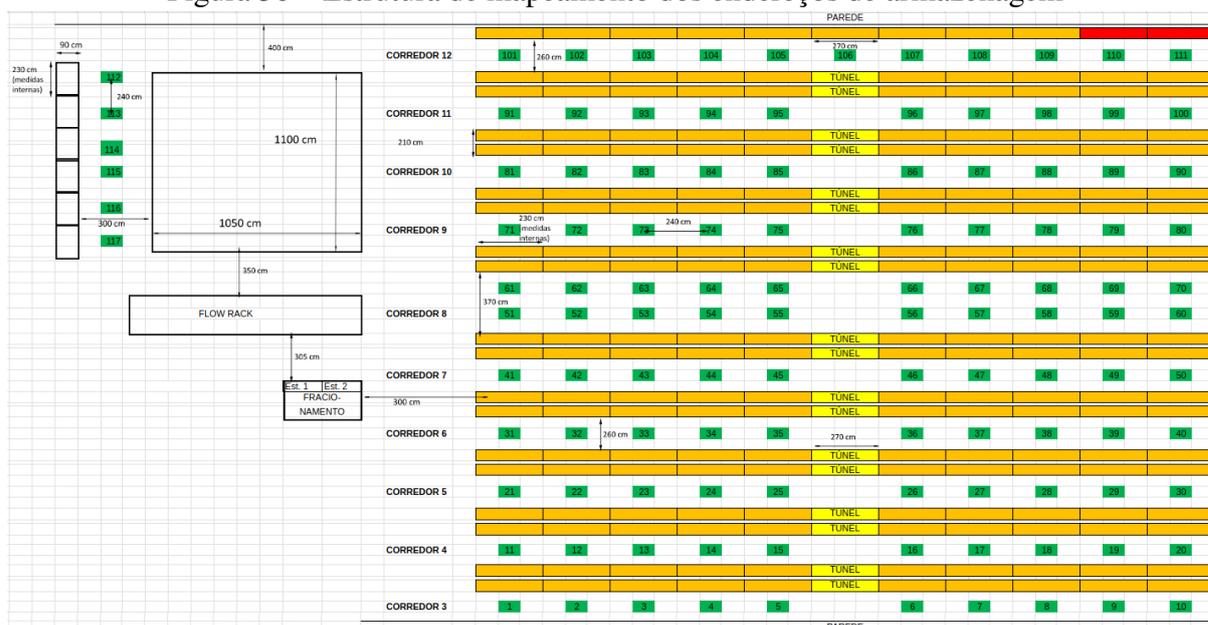
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
20	CORREDOR 5														
21	PC	Prédios	Endereços	PC	Prédios	Endereços	PC	Prédios	Endereços	PC	Prédios	Endereços	PC	Prédios	Endereços
22	21	5 - 2 - 1	101 a 104	22	5 - 6 - 1	101 a 104	23	5 - 10 - 1	101 a 104	24	5 - 14 - 1	101 a 104	25	5 - 18 - 1	101 a 104
23		5 - 4 - 1	201 a 204		5 - 8 - 1	201 a 204		5 - 12 - 1	201 a 204		5 - 16 - 1	201 a 204		5 - 20 - 1	201 a 204
24			301 a 304			301 a 304			301 a 304			301 a 304			301 a 304
25			401 a 404			401 a 404			401 a 404			401 a 404			401 a 404
26		Prédios	Endereços		Prédios	Endereços		Prédios	Endereços		Prédios	Endereços		Prédios	Endereços
27		5 - 1 - 1	101 a 104		5 - 5 - 1	101 a 104		5 - 9 - 1	101 a 104		5 - 13 - 1	101 a 104		5 - 17 - 1	101 a 104
28		5 - 3 - 1	201 a 204		5 - 7 - 1	201 a 204		5 - 11 - 1	201 a 204		5 - 15 - 1	201 a 204		5 - 19 - 1	201 a 204
29			301 a 304			301 a 304			301 a 304			301 a 304			301 a 304
30			401 a 404			401 a 404			401 a 404			401 a 404			401 a 404
31															
32	CORREDOR 6														
33	PC	Prédios	Endereços	PC	Prédios	Endereços	PC	Prédios	Endereços	PC	Prédios	Endereços	PC	Prédios	Endereços
34	31	6 - 2 - 1	101 a 104	32	6 - 6 - 1	101 a 104	33	6 - 10 - 1	101 a 104	34	6 - 14 - 1	101 a 104	35	6 - 18 - 1	101 a 104
35		6 - 4 - 1	201 a 204		6 - 8 - 1	201 a 204		6 - 12 - 1	201 a 204		6 - 16 - 1	201 a 204		6 - 20 - 1	201 a 204
36			301 a 304			301 a 304			301 a 304			301 a 304			301 a 304
37			401 a 404			401 a 404			401 a 404			401 a 404			401 a 404
38		Prédios	Endereços		Prédios	Endereços		Prédios	Endereços		Prédios	Endereços		Prédios	Endereços
39		6 - 1 - 1	101 a 104		6 - 5 - 1	101 a 104		6 - 9 - 1	101 a 104		6 - 13 - 1	101 a 104		6 - 17 - 1	101 a 104
40		6 - 3 - 1	201 a 204		6 - 7 - 1	201 a 204		6 - 11 - 1	201 a 204		6 - 15 - 1	201 a 204		6 - 19 - 1	201 a 204
41			301 a 304			301 a 304			301 a 304			301 a 304			301 a 304
42			401 a 404			401 a 404			401 a 404			401 a 404			401 a 404
43															

Fonte: O Autor (2024)

Em seguida, foram realizadas medições das dimensões dos espaços e das estruturas físicas do armazém utilizado no estudo. Essas medidas serviram de base para a criação de uma nova planilha, detalhando com precisão a disposição física dos endereços, pontos de coleta e áreas relevantes para o mapeamento físico. Com base nessas medidas, determinou-se uma escala que permitiu a criação de uma representação precisa do *layout* do armazém, onde cada célula da planilha representa uma proporção real, conforme mostrado na Figura 50.

Para transpor as informações coletadas para um sistema computacional, os dados foram reorganizados em um formato mais simplificado, agrupando os elementos em três categorias principais: endereços de armazenamento, pontos de coleta e obstáculos. Cada elemento foi associado a coordenadas cartesianas (x, y) que indicam sua localização no *layout* virtual do armazém. Essas coordenadas permitiram criar um mapa bidimensional detalhado, onde os elementos re-

Figura 50 – Estrutura de mapeamento dos endereços de armazenagem



Fonte: O Autor (2024)

gistrados nas planilhas foram posicionados com base em suas localizações reais.

No modelo computacional, cada célula da matriz bidimensional representa um elemento do armazém, como prédios, obstáculos ou pontos de coleta. Por exemplo, se a planilha indica dois prédios localizados nas coordenadas (10, 10) e (11, 10), esses elementos são inseridos lado a lado na matriz, refletindo fielmente sua disposição no ambiente físico.

Com base neste mapa 2D, foi desenvolvido um modelo virtual capaz de calcular, com uma boa precisão, informações que seriam inviáveis financeira e operacionalmente de se coletar manualmente. Isso inclui, por exemplo, o cálculo das distâncias entre um ponto de coleta específico, e todos os outros pontos existentes no armazém, que somando, resultam em 118 medidas. Multiplicando pela necessidade de encontrar as distâncias entre todos os pontos, esse número chega a 13.924 valores.

APÊNDICE L – CONCEPÇÃO DOS PONTOS DE COLETA

Para que os robôs consigam navegar e se localizar durante as tarefas de *picking* no armazém, é necessário que eles reconheçam os locais em que devem parar a cada posição de coleta solicitada, bem como saber os pontos de depósito para destinar os produtos coletados. Portanto, essas informações precisam estar integradas e ser consideradas no mapeamento do ambiente que será utilizado pelos robôs. É importante destacar que, nem todo endereço de *picking* ou prédio¹ corresponde diretamente a um ponto de parada específico para o robô. Além disso, a identificação desses pontos pelos robôs varia de acordo com o tipo de tecnologia de navegação empregada por ele (Seção 2.2).

Apesar de ser contraintuitivo, utilizar pontos de parada correspondentes a cada endereço de coleta, além de aumentar a complexidade, não garante melhorias de desempenho. Isso se deve ao fato de que, para os robôs, realizar movimentações frequentes para distâncias curtas pode aumentar o tempo de execução das tarefas, devido ao tempo consumido não apenas no deslocamento em si, mas também nos processos de início e conclusão de cada movimento. Além disso, esses deslocamentos podem reduzir a sua eficiência energética, uma vez que o consumo de energia para iniciar e parar os AGVs ou AMRs pode não ser efetivamente compensado pela curta distância percorrida.

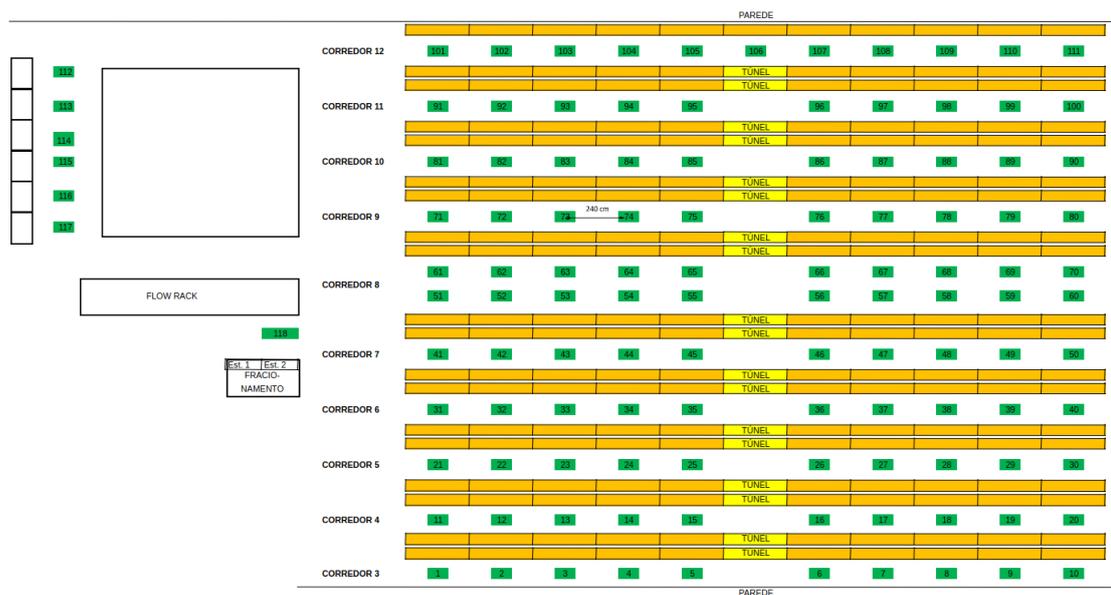
Para resolver essa questão, este estudo propôs e implementou o uso de pontos de coleta (PC) centralizados como solução. Esses pontos agrupam em módulos diversos endereços de *picking* próximos que foram posicionados estrategicamente para balancear a distância percorrida pelos robôs e garantir fácil acesso aos endereços vinculados. Essa abordagem tem como objetivo reduzir movimentações curtas e frequentes, aprimorando tanto o tempo de execução das pequenas movimentações quanto o consumo energético. Além disso, a introdução dos pontos de coleta não apenas simplifica o mapeamento e a navegação dos robôs, mas também reduz significativamente a complexidade do sistema de otimização, reduzindo consideravelmente o tamanho das instâncias.

Nesse sentido, considerando o cenário real utilizado para as experimentações, o mapeamento dos pontos de coleta para navegação dos AMRs foi planejado e composto por módulos distando entre si, 2,4 metros. Para considerar essas operações no modelo de otimização, foi necessário realizar um mapeamento lógico que espelhasse a estrutura física do armazém, levando em consideração a disposição dos pontos de coleta. Esses pontos foram posicionados próximos aos endereços, respeitando o espaçamento de 2,4 metros e permitindo o atendimento eficiente de até dois prédios ao mesmo tempo.

Na Figura 51, pode-se visualizar o resultado deste mapeamento. Destacadas em laranja estão as estruturas porta-paletes e em verde estão os pontos de coleta criados. A maior parte dos

¹ Refere-se à estrutura de armazenamento, seja um porta *pallet* ou prateleira que vai de uma longarina vertical até a outra. Para mais informações, consultar Apêndice H

Figura 51 – Pontos de coleta (em verde) posicionados no armazém



Fonte: O Autor (2024)

pontos de coleta atenderá um grupo de endereços que se estende a dois prédios. Por exemplo, o ponto 101, situado no início do corredor 12, abrange tanto o prédio imediatamente acima quanto abaixo, portanto, engloba toda a faixa de endereços pertencentes a essas duas estruturas. Existem casos, como no corredor 8, em que, devido a maior largura do corredor, cada ponto de coleta atende apenas um prédio, de modo a não causar prejuízos no deslocamento dos operadores e empilhadeiras.