



Eficiência da atividade de order picking orientada por algoritmo baseado na heurística s-shape: Uma investigação exploratória

Cleuson Sousa Santos¹, Marcelo Henrique Oliveira Henklain¹, Felipe Leite Lobo¹, Yaritza Barreto¹

¹Departamento de Ciência da Computação – Universidade Federal de Roraima (UFRR)
– Boa Vista – RR – Brasil

cleusonss@gmail.com, {marcelo.henklain, felipe.lobo,
yaritza.barreto}@ufrr.br

Abstract. *The order picking (OP) process consists of searching for and separating items in distribution centers (DCs), being the most impactful factor in operational costs. Especially small DCs need strategies to optimize this process, as they lack the resources to automate it. To address this issue, we investigated the impact of assisting human OP through an algorithm that implements the S-Shape heuristic, using a multiple baseline design. Two experienced workers participated, with their task being to pick lists of medications as quickly as possible. At times, the participant could define the picking route, while at other times, they had to follow the route determined by the algorithm. An increase in OP efficiency was observed when the route was guided by the algorithm compared to the route determined by the participant.*

Resumo. *O processo de order picking (OP) consiste na busca e separação de itens em centros de distribuição (CD), sendo o que mais impacta nos custos operacionais. Especialmente CDs pequenos precisam de estratégias para otimizar esse processo, pois não possuem recursos para automatizá-lo. Para lidar com esse problema, investigamos com um delineamento de linha de base múltipla o impacto do auxílio ao OP humano por meio de algoritmo que implementa a heurística S-Shape. Participaram dois trabalhadores experientes, cuja tarefa foi buscar listas de medicamentos no menor tempo possível. Ora o participante podia definir a rota de busca, ora devia seguir a rota determinada pelo algoritmo. Verificou-se aumento da eficiência do OP com rota orientada pelo algoritmo em comparação com a rota determinada pelo participante.*

1. Introdução

O processo de *order picking* (OP) consiste na busca e separação de itens que compõem um pedido, sendo tarefa fundamental em quase todas as cadeias de suprimento (Winkelhaus et al., 2021). Essa atividade é a que mais impacta nos custos operacionais de um Centro de Distribuição (CD), podendo corresponder a mais da metade desses custos, o que, seguramente, afeta o preço final dos produtos para o consumidor (Cano, 2017; Masae, 2019b). Por essa razão, a indústria busca alternativas tecnológicas que possam otimizar esse processo.

Segundo Urzúa (2019), embora as pequenas e médias empresas de distribuição de diferentes setores da economia empreendam esforços para melhorar o tempo de resposta do OP, o desempenho ainda não é competitivo comparado a grandes empresas de



comércio eletrônico, a exemplo da Amazon, que crescem rapidamente, possuem alto grau de automatização de seus processos e, assim, ameaçam a estabilidade econômica dos pequenos e médios negócios. Por isso, os pequenos e médios centros de distribuição buscam alternativas para otimizar o atendimento de pedidos, de modo a atingir padrões de competitividade de nível mundial, buscando obter sucesso mesmo sem a automação integral de seus processos. Esses centros de distribuição precisam, portanto, contar com a colaboração entre máquinas e humanos, já que o custo de automatizar todo o processo inviabilizaria o próprio negócio (Winkelhaus et al., 2021).

Para lidar com esse problema e atuar no auxílio aos pequenos e médios negócios, esta pesquisa investigou a eficiência da atividade de *order picking*, com e sem aplicação de algoritmo que implementa a heurística *S-Shape*, para atender pedidos dentro de um centro de distribuição de medicamentos. Desse modo, implementamos esse algoritmo e fizemos a comparação de dois cenários: no primeiro, o separador definiu a própria ordem de busca dos itens do pedido; no segundo cenário, o algoritmo determinou a ordem de busca dos medicamentos a serem separados.

Este trabalho está organizado da seguinte forma: começamos pela fundamentação teórica, na qual apresentamos conceitos pertinentes para a compreensão da pesquisa. Na seção de trabalhos relacionados abordamos estudos que comparam o impacto das políticas de separação sobre o desempenho da tarefa de atender os pedidos dentro de um centro de distribuição. No método, abordamos como nossa solução computacional foi implementada e testada a partir de um delineamento de linha de base múltipla. Apresentamos, então, os nossos achados na seção de resultados e discussão e, na última seção, finalizamos este artigo com a conclusão.

2. Fundamentação Teórica

2.1. *Order picking*

A expressão em inglês *order picking* (OP) pode ser definida como o processo de coleta de itens para atender um pedido no contexto de um centro de distribuição ou estoque (Winkelhaus et al., 2021). Em português, essa expressão pode ser traduzida como separação de pedidos. Neste trabalho adotaremos o termo em inglês por ser o padrão na literatura científica.

Em razão do seu impacto nos custos operacionais, o processo de OP tem sido sistematicamente estudado por diversas ciências, envolvendo pesquisas em áreas como engenharia de produção, estatística, matemática e ciência da computação, sendo um campo vital de pesquisa para a indústria. Dados levantados por Winkelhaus et al. (2021) demonstram a importância da pesquisa sobre OP ao evidenciarem, nos últimos 30 anos, o aumento exponencial de artigos indexados na base de dados *Scopus*, com o termo de busca *order picking* no título.

Na literatura internacional, o processo de OP é dividido e analisado em etapas, são elas: (1) Planejamento e preparação da coleta, (2) Viagem e transporte, (3) Busca do local em que o item será coletado, (4) Coleta dos itens, e (5) Confirmação da coleta. Assim, cada pesquisa enfatiza uma ou mais dessas etapas de OP. Neste trabalho, vamos investigar especificamente as etapas de (2) Viagem e transporte, (3) Busca do local do item a ser coletado e de (4) Coleta dos itens, que são aspectos cruciais do processo de OP. Segundo Urzúa (2019), essas etapas, no que concerne à comparação da eficácia de



diferentes políticas de *picking*, têm sido o foco principal das pesquisas internacionais sobre OP.

Seguindo essa linha de investigação, sabe-se que a maioria das empresas ainda faz uso do OP manual, o que pode ser justificado devido ao grande investimento necessário para a implantação de um sistema automatizado (Masae, 2019a). Por isso, existe um número crescente de pesquisas apontando possibilidades de automatizar parte da tarefa de separação de pedidos ou dar suporte aos trabalhadores por meio da tecnologia, por exemplo, pela determinação da rota de busca ou ordem dos itens que precisam ser separados (ex.: Elbert et al., 2017; Ozden et al., 2020). Diante disso, são identificadas algumas vantagens dessa modalidade de OP, tais como a possibilidade que o trabalhador humano tem de lidar melhor com a variabilidade em termos de forma e tamanho dos itens a serem coletados e com a variabilidade da demanda, decorrente da sazonalidade (Urzúa, 2019; Winkelhaus et al., 2021). Essas vantagens aumentam o estímulo para pesquisas que busquem otimizar esse processo de OP.

2.2. Políticas de rotas de coleta

Com o intuito de aumentar a eficiência da operação, os centros de distribuição (CD) implementam políticas de rotas com os seguintes objetivos: (1) minimizar a distância percorrida pelos separadores durante a coleta dos itens dos pedidos; (2) diminuir a fadiga dos separadores; e (3) o tempo de realização das atividades. De acordo com Masae et al. (2019a), como o tempo de deslocamento é responsável por mais de 50% do tempo total de separação do pedido, o problema das rotas de coleta tem recebido atenção da comunidade científica (ver, por exemplo, Grosse et al., 2017; Van Gils et al., 2018).

Dijkstra (2017) afirma que dentre as políticas de rota, é comum aplicar os métodos (a) *S-Shape*: o separador entra por uma extremidade do corredor de separação e sai na outra, passando por todas as ruas que contêm coletas antes de retornar ao ponto inicial; (b) *Return*: o separador entra e sai de cada corredor de separação contendo, pelo menos, um local de separação do mesmo lado; (c) *Midpoint*: o separador entra em um corredor de separação apenas até o ponto médio e retorna para deixar o corredor de separação pela mesma extremidade; (d) *Largest gap*: o separador entra em um corredor de separação apenas até o início do maior intervalo e retorna para deixar o corredor de separação pela mesma extremidade na qual entrou. Para cada uma dessas políticas é possível trabalhar com estratégias de otimização.

Quando se trata de pequenos e médios centros de distribuição, nos quais atuam separadores humanos, é possível identificar na literatura que uma das políticas mais promissoras de rota é a heurística *S-Shape* (Masae et al., 2019b; Valle et al., 2017). O motivo disso é a simplicidade e flexibilidade dessa heurística, que foi originalmente desenvolvida para CD de bloco único, ou seja, pequeno, e com modificações também aplicadas a CD de bloco múltiplo, isto é, grande (Cano et al., 2017).

2.3. Otimização e Métricas de Performance

Otimização é um tema muito discutido em várias áreas do conhecimento. Masae et al. (2019a) dissertam sobre esse tema indicando que existem três tipos de algoritmos propostos na literatura para a resolução do problema de rotas: (1) Algoritmos exatos, que sempre acham uma solução ótima (ex.: Problema do Caixeiro Viajante); (2) Heurísticas, que são algoritmos dependentes de problemas criados de acordo com suas especificações



(ex.: *Midpoint* etc.); e (3) Meta-heurísticas, que são algoritmos independentes de problemas de alto nível, que fornecem um conjunto de diretrizes ou estratégias para encontrar uma solução aproximada para o problema (ex.: Colmeia de Abelha, Colônia de Formigas etc.). Em todas as técnicas de otimização, faz-se necessário delimitar parâmetros numéricos acerca da performance esperada.

Valle et al. (2017) afirmam que, na prática, o problema de políticas de rotas em um CD é resolvido principalmente usando heurísticas e citam como principais razões pelas quais elas são usadas (no lugar de algoritmos ótimos) as seguintes: (1) Nem todo armazém é capaz de utilizar um algoritmo ideal, devido ao longo tempo de computação exigido para produzir uma solução ótima; (2) Rotas ótimas podem parecer ilógicas para os separadores e, como resultado, eles optam por se desviar das rotas computadas que são atribuídas a eles; e (3) um algoritmo ótimo não leva em consideração as condições do ambiente, como congestionamento do corredor, ao passo que é possível evitar ou reduzir esse tipo de problema com heurísticas de roteamento.

Desse modo, métricas de performance são aplicadas para avaliar em que medida um algoritmo foi bem-sucedido em sua tarefa. Sobre esse tema, Van Gils et al. (2018) avaliam que o tempo de execução da tarefa de OP é uma dimensão fundamental do processo a ser otimizado. Em termos de tempo, importa considerar, principalmente, o tempo de viagem e o tempo de pesquisa e localização de itens. Com efeito, Van Gils et al. (2018) afirmam que a maioria das publicações sobre métricas de desempenho de *order picking* tem como objetivo otimizar o tempo despendido. Por fim, pesquisas que consideram indicadores de desempenho relacionados ao tempo, incluem pelo menos o tempo de viagem, supondo que outros componentes do tempo sejam constantes (Elbert et al., 2017; Urzúa et al., 2019).

Nessa perspectiva, adotaremos neste estudo a variável tempo de busca de itens como nosso foco. Utilizaremos a política de rota *S-Shape*, o que se deve ao fato de a literatura apontá-la como a que implementa uma estratégia mais simples para os separadores, e por isso de fácil assimilação e uso (Cano et al., 2017; Redmer, 2020; Dijkstra, 2017).

2.4 Delineamento de linha de base múltipla

O delineamento de linha de base múltipla é um tipo de procedimento de coleta de dados adequado para quando dispomos de poucos participantes e precisamos entender melhor de que modo uma variável independente impacta uma variável dependente para uma pessoa específica. Por ser um delineamento de sujeito único, o controle do experimento não depende de um grupo, mas de medidas feitas acerca do comportamento de cada indivíduo antes de uma intervenção e o que ocorre com o comportamento, conforme a variável independente é manipulada (Sampaio et al., 2008).

2.5. Tecnologias para o desenvolvimento de software

Para atender às necessidades específicas da pesquisa, foi criada uma interface de programação de aplicações (API). O desenvolvimento da API envolveu a utilização de múltiplas tecnologias que serão explicadas nas subseções a seguir. Essas tecnologias permitiram o desenvolvimento do que, no método, apresentamos como solução computacional.



2.5.1. JavaScript

JavaScript (JS) é uma linguagem de *script* multiplataforma, inicialmente projetada para interagir com elementos de páginas da web (Mozilla et al., 2022). Os mecanismos JavaScript modernos podem ser executados tanto em navegadores quanto em servidores, sendo listada no site < <https://midnight.github.io> > como uma das linguagens de programação com maior número de repositórios no GitHub, o que evidencia a sua popularidade (Mozilla et al. 2022; Pano, 2018; Sun, 2017). Tem no Node.js sua versão mais avançada do lado do servidor, que permite adicionar mais funcionalidades, como conexões com bancos de dados. Esta implementação dispensa o uso de navegador (Mozilla et al., 2017).

Em comparação de performance com Java, C++ e Python, Abdulkareem (2021) afirma que o JS mantém performance equiparada, é mais simples e mais performática que Java, e pouco menos performática que Python e C++. Considerando as características dessa linguagem e o fato de que o problema que precisamos resolver possui como requisito a utilização de uma linguagem que garanta performance e permita ser utilizada tanto no desenvolvimento do servidor quanto do cliente, decidimos em favor do uso do JS para implementação do algoritmo baseado na política de rota *S-Shape*.

2.5.2. Node.js

A OpenJS Foundation (2022) define o Node.js como um ambiente de tempo de execução JavaScript de código aberto e multiplataforma. É uma ferramenta popular para qualquer tipo de projeto. A eficiência do Node.js é garantida devido ao fato de ele executar o mecanismo JavaScript V8, o núcleo do Google Chrome, fora do navegador. Um aplicativo Node.js é executado em um único processo, sem criar *threads* para cada solicitação. Apresenta como vantagem a capacidade de escrever o código do lado do servidor, com a mesma tecnologia que se escreve código para o cliente, portanto, eliminando a necessidade de aprender uma linguagem completamente diferente. Dentre as razões para a sua popularidade, podem ser citadas: (1) o gerenciamento eficiente de entradas e saídas e (2) o maior ecossistema de bibliotecas de código aberto do mundo, o *Node Package Manager* ou NPM (OpenJS Foundation, 2022). Considerando as suas vantagens e o fato de que a solução do problema deve ser capaz de ser executada em qualquer plataforma, decidimos usar o Node.js.

2.5.3. Visual Studio Code

O Visual Studio Code (VSC) é um editor de código-fonte leve, mas poderoso, que é executado em área de trabalho, multiplataforma, com suporte integrado para JavaScript e Node.js, possuindo um rico ecossistema de extensões (Microsoft, 2022). Tendo por base essa integração de tecnologias, avaliamos que essa ferramenta facilitaria o nosso trabalho, tendo sido escolhida por essa razão.

2.5.4. MongoDB e Mongoose

O MongoDB é um banco de dados orientado a documentos, projetado para facilitar o desenvolvimento e a escalabilidade dos aplicativos. Um registro no MongoDB é um documento, que é uma estrutura de dados composta por pares de campo e valor. Os documentos MongoDB são semelhantes a objetos JSON. Os valores podem incluir outros documentos, matrizes e matrizes de documentos (MongoDB, 2022a). Para simplificar a



estruturação e o acesso aos dados persistidos no MongoDB, é utilizada a biblioteca de terceiros chamada Mongoose. Como motivação para usar o Mongoose está o fato de ele forçar um esquema semirrígido desde o início do projeto, em que o desenvolvedor deve definir um esquema e modelo de dados (Hall, 2022). Optamos pelo uso dessa ferramenta porque a combinação dessas duas tecnologias aumenta a produtividade do desenvolvimento, visto que o formato do dado do banco de dados é o mesmo utilizado pela linguagem de programação.

3. Trabalhos Relacionados

Considerando o objetivo deste estudo, decidimos realizar uma breve revisão da literatura, para identificar pesquisas que analisam o desempenho de algoritmos de *order picking* e fazem comparações entre algoritmos e, ainda, entre o desempenho do trabalho de separação executado por trabalhadores humanos com e sem auxílio de algoritmos. Esta revisão foi realizada consultando o Portal de Periódicos da CAPES. Para restringir a busca a artigos que fossem convergentes com o objetivo proposto, foram adotadas algumas condições acompanhando a *string* de busca: Título com termos exatos: *order picking* AND qualquer campo CONTÉM *human factor* AND qualquer campo CONTÉM *route policies*. Tipo de Material: Todos os itens; Idioma: Qualquer Idioma; Data de publicação: Últimos 5 anos. A busca realizada retornou 11 estudos, cujos títulos e resumos foram lidos na íntegra.

A partir dessas leituras, foram selecionados os estudos mais aderentes em relação à proposta do presente trabalho. Desta maneira, escolhemos artigos que realizam comparações diretas entre desempenho de algoritmos e artigos que comparam o desempenho do ser humano auxiliado por algoritmos em relação ao trabalho do separador de CD sem auxílio de um algoritmo. A partir dos trabalhos encontrados, verificamos que são recorrentes na literatura os estudos que utilizam simulações computacionais, sendo uma importante lacuna a condução de estudos que coletem dados com seres humanos em cenários reais de *order picking*. A seguir, descrevemos os estudos selecionados.

Urzúa et al. (2019) conduziram estudo cujo objetivo foi avaliar o impacto de três estratégias de *order picking*, em dois cenários, sobre o tempo de atendimento de um pedido em um CD real de ferragens, em que utilizaram 154.325 *picks* atendidos como entrada para um modelo de simulação. Na operacionalização do OP real foram utilizados 46 separadores humanos e a central de armazenamento foi dividida em zonas de *picking*, cada uma com um conjunto específico de características, levando em consideração o tamanho dos itens a serem separados. Dentre as três zonas de *picking*, uma foi designada como zona de *picking* automatizada, em que humanos eram auxiliados por robôs, e as outras duas zonas foram compostas apenas por trabalhadores humanos. Tal medida foi adotada para viabilizar a comparação de eficiência do algoritmo entre as três zonas. Como resultado, percebeu-se que houve redução de 55% do tempo médio de atendimento por pedido após a implementação da estratégia automatizada, que combinava robôs e humanos. Nesse caso, as estratégias de OP automatizado contribuíram para uma força de trabalho mais produtiva, observando-se uma melhora média de 69,25% na produtividade.

Outro estudo que examinou o melhor tipo de estratégia de OP foi o de Ozden et al. (2020). Esses pesquisadores propuseram uma modelagem do percurso da viagem para busca de um pedido que estima o menor percurso, levando em consideração o comportamento real do trabalhador e a sua comparação com algoritmos que determinam



o menor caminho, utilizando ângulos retos. Os autores concluíram que há diferenças entre a operação manual e a automatizada. Essa diferença consiste no modo que humanos e máquinas percorrem uma trajetória, e consideraram que os efeitos do “corte dos cantos” de corredores em ambientes automatizados são reduzidos porque os veículos guiados autonomamente seguem caminhos prefixados e determinados pelos algoritmos. Por isso representam bem as métricas definidas nos CDs, enquanto os trabalhadores, além de “cortarem os cantos”, não mudam de direção em ângulo reto. Ozden et al. afirmam que o modelo tradicional de distâncias, com base em centros de corredor, pressupõe que os trabalhadores andam nos centros dos corredores e mudam de direção em ângulo reto, enquanto o modelo alternativo proposto pressupõe que farão esforços razoáveis para cortar os cantos e, portanto, tomar os caminhos mais curtos para seus destinos, mostrando que o método de centros de corredor superestima a distância que um trabalhador racional andaria. Esse estudo demonstrou sua importância ao identificar que o rendimento para um determinado número de separadores deve ser maior do que o previsto pelo modelo tradicional de simulação em virtude de os trabalhadores percorrerem caminhos mais curtos do que o suposto.

Por fim, com o intuito de avaliar o impacto do comportamento humano sobre o processo de *order picking*, Elbert et al. (2017) conduziram estudo cujo objetivo foi comparar a distância percorrida por um profissional separador em duas situações: (1) na rota preestabelecida por um algoritmo de separação e (2) em uma condição com desvio dessa rota preestabelecida. Para dar conta desse objetivo, foi desenvolvida uma simulação de computador que mediu a distância percorrida pelo separador em uma rota e identificou a menor rota em um conjunto de possibilidades testadas. Neste estudo, verificou-se aumento do tempo de separação quando o separador desvia da rota determinada pelo algoritmo, ou seja, ocorre piora no desempenho da separação. Dessa forma, o algoritmo de separação se mostrou mais eficiente comparativamente aos métodos de OP propostos pelos funcionários.

Os estudos apresentados revelam, por um lado, que muitas variáveis determinam o comportamento humano de *order picking*, não sendo uma tarefa simples simular corretamente o que um trabalhador humano faz, de modo a comparar o seu desempenho com o de uma máquina. Por outro lado, esses mesmos estudos sugerem a possibilidade de ganhos de eficiência no processo de *order picking* realizado por humanos, quando eles são orientados por um algoritmo. Observamos também que poucos são os estudos que de fato comparam a eficiência da atividade de *order picking*, realizada por humanos, nas condições com e sem apoio de um algoritmo, que determine a ordem dos itens a serem coletados. Além disso, poucas são as pesquisas que realizam esse tipo de comparação em um cenário real, mas com um grau rigoroso de controle de variáveis que afetam o comportamento individual de *order picking*. Nesta pesquisa, buscamos lidar com essas lacunas na literatura. Por isso, o nosso objetivo foi investigar a eficiência da atividade humana na tarefa de *order picking* com o auxílio de algoritmo de determinação de melhor rota de busca.

4. Método

4.1. Solução computacional

A pesquisa, realizada em um centro de distribuição de medicamentos, localizado na cidade de Boa Vista/RR, foi organizada em dois momentos: (1) o separador realizava o

order picking de uma lista de pedidos utilizando sua própria estratégia e (2) realizava o *order picking* seguindo a ordem de coleta determinada pela heurística implementada, apresentada para o separador a partir de uma lista impressa com itens ordenados na sequência que deveriam ser coletados.

Desse modo, buscamos mensurar o tempo de busca para completar o atendimento de um pedido, para, assim, avaliar se ocorreria redução do tempo gasto para completar a tarefa na condição de *order picking* orientado por algoritmo. Para melhor compreensão dos dados obtidos, iremos tomar como foco a viagem do separador ao longo dos corredores em que foi implementada a política de rota *S-Shape*, que consiste em atravessar um corredor inteiro, caso exista algum item a ser coletado. Corredores que não possuem itens na lista de separação não são visitados. Para essa heurística, o separador sai de um local determinado, coleta todos os produtos da lista e retorna ao ponto inicial, não sendo obrigatório ir até o final do último corredor. A Figura 1 exibe um exemplo de rota percorrida obedecendo a política de rotas *S-Shape*.

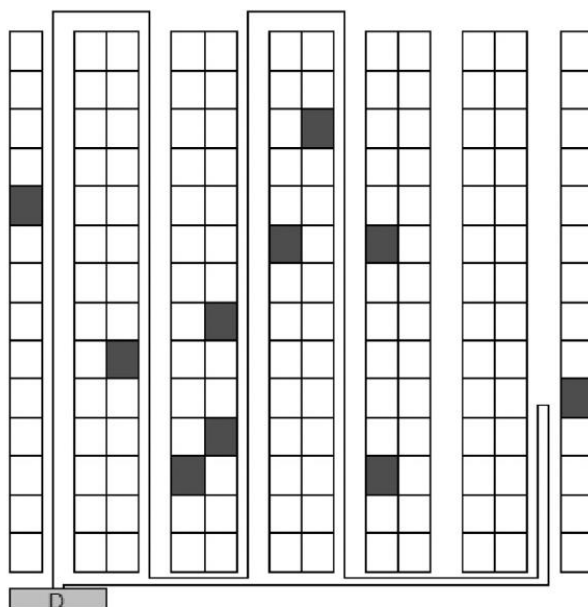


Figura 1. Exemplo de política de rota *S-Shape*.

Nota. Os quadros pintados indicam locais de *picking*.

Fonte: Dijkstra (2017).

A Figura 2 exibe, em *JavaScript*, o código fonte do algoritmo que implementa a heurística *S-Shape*. Esse programa foi disponibilizado no GitHub (<https://github.com/cleusonss/sshape-heuristic>) e está disponível para uso segundo a licença *Apache 2.0*.

É importante ressaltar que a política de rotas denominada *S-Shape* é uma heurística. Por isso, os termos "política de rotas *S-Shape*", "heurística *S-Shape*", "método *S-Shape*" e "algoritmo *S-Shape*" são sinônimos, visto que todos se referem ao mesmo algoritmo que visa obter uma solução para o problema de *order picking*. Neste trabalho, o algoritmo que implementa a heurística *S-Shape* foi escolhido por ser extensamente documentado na literatura, além de ser indicado como eficiente para esse tipo de



operação, bem como por possuir flexibilidade para se adaptar a diferentes tipos de armazéns e ser facilmente compreendido pelos separadores humanos.

```
// 1. Recebe uma ordem com lista de produtos
let order = produtos;

// 2. Ordena a lista
order = order.sort();

// 3. Busca o primeiro corredor
let c = order[0].corredor;

// 4. Caminha até o final do primeiro corredor
let sshape = order.filter(produto => produto.corredor === i);

// 5. Variável de decisão de retorno
let retorno = true;

// 6. Caminha pelos próximos corredores
for (c = c+1; c <= ULTIMO_CORREDOR; c++){

// 7. Caminha pelo corredor
    let corredor = order.filter(produto => produto.corredor ===
        i);

// 8. Verifica se o corredor tem produto
    if (corredor){

// 9. Decisão de retorno
        if (retorno){

// 10. Armazena os valores e inverte o retorno
            sshape = sshape.concat (corredor.reverse());
            retorno = false;

        }else{

// 11. Armazena os valores e inverte o retorno
            sshape = sshape.concat (corredor);
            retorno = true;

        }
    }

// 12. Retorna a rota que o separador deve seguir
return sshape;
```

Figura 2. Algoritmo em JavaScript que implementa a heurística S-Shape.

Fonte: Autor (2023).

Para tornar o uso do algoritmo viável, poder fazer a entrada de dados para processamento e, conseqüentemente, obter a rota de separação como saída, foi desenvolvida uma API, que permitiu que o algoritmo fosse disponibilizado como serviço para outras aplicações utilizarem a sua funcionalidade. Essa API foi desenvolvida utilizando a linguagem de programação JavaScript, com Node.js e NPM para gerenciar



pacotes de bibliotecas. Por fim, para o armazenamento de dados de produtos e ordem de separação, foi utilizado o banco de dados MongoDB, hospedado no serviço em nuvem MongoDB Atlas.

A API foi implementada com os seguintes itens: (1) *core* da heurística *S-Shape*, que ao receber uma lista de produtos a ordena seguindo essa heurística, utilizando-se das informações de corredor e posição do produto; (2) *controller* para produtos e ordens de separação, que contém as regras de negócios sobre operações, produto e ordens de separação; (3) modelos para produtos e ordens de separação contendo os esquemas e modelos de dados do produto e das ordens de separação; (4) serviço de conexão de dados para persistência no banco; (5) rotas que definem os *endpoints* de conexão para entrada e saída de dados; e (6) serviço http, para permitir a comunicação entre aplicações clientes e a API. Na Tabela 1 estão elencadas as ferramentas e tecnologias adotadas no desenvolvimento da aplicação, utilizando a arquitetura de *software* baseada em API, amplamente utilizada para construir sistemas distribuídos e escaláveis, em que várias aplicações podem se comunicar por meio de uma interface comum. Com essa abordagem, é possível dividir as responsabilidades da aplicação em diferentes camadas, facilitando a manutenção e evolução do sistema ao longo do tempo.

Tabela 1. Ferramentas e tecnologias utilizadas para no desenvolvimento da aplicação.

Ferramenta	Versão	Referência	Finalidade
JavaScript	ECMA Script 2022	https://www.javascript.com/	Linguagem multiparadigma para desenvolvimento web e servidor.
Visual Studio Code	1.74.1	https://code.visualstudio.com/	Editor de Código Fonte
Node.js	16.15.1	https://nodejs.org	Plataforma para construção do servidor
MongoDB Atlas	6.0	https://cloud.mongodb.com/	Servidor de banco de dados.
Mongoose	6.6.5	https://mongoosejs.com/	Biblioteca para facilitar a interação com o banco de dados
Git	2.37.1	https://git-scm.com/	Versionamento do Código
Postman	10.6.7	https://www.postman.com/	Cliente para testar API

4.2. Avaliação experimental

Para que seja possível determinar se o algoritmo proposto na seção de solução computacional resulta em melhor desempenho humano na tarefa de *order picking* (OP), em comparação com o trabalho de separação baseado apenas nos critérios do trabalhador, adotamos um delineamento de linha de base múltipla (LBM) com comparação entre participantes e com uma reversão. A variável dependente (VD) foi o tempo utilizado para separar um conjunto de medicamentos e a variável independente (VI) foi a determinação da rota de coleta de medicamentos pelo algoritmo combinado com o endereçamento de itens. A reversão dentro do procedimento de LBM consiste na retirada da VI

acompanhada pela mensuração de seus efeitos sobre a VD, seguida pela reinserção da VI e nova medida da VD.

4.2.1. Participantes

Participaram do estudo dois separadores, P1 e P2, ambos do sexo masculino, com experiência de trabalho em relação à tarefa de *order picking*. P1 tinha 28 anos de idade e trabalhava na empresa há 01 ano e 02 meses. P2 tinha 34 anos de idade e trabalhava na empresa há 01 ano e 05 meses.

4.2.2. Local e materiais

A coleta de dados foi realizada em um armazém de medicamentos, medindo 262 m², conforme a Figura 3. O armazém era formado por seis corredores horizontais e três verticais, onde somente os corredores horizontais estavam disponíveis para o armazenamento, e os corredores verticais eram projetados para permitir o trânsito dos separadores. Os corredores horizontais continham 08 seções de estantes com 07 prateleiras (níveis), e em cada prateleira havia seis posições de armazenamento. Com essa configuração de corredores e estantes, era possível armazenar 336 produtos diferentes.

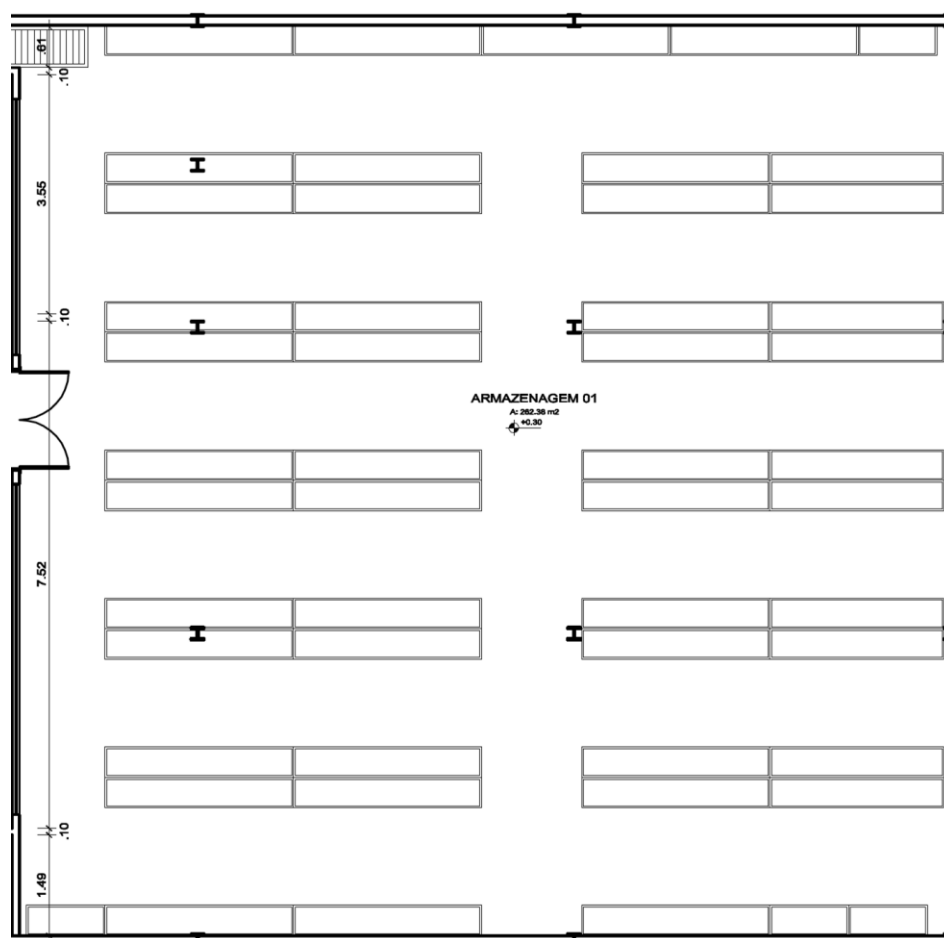


Figura 3. Planta baixa do armazém de medicamentos.



Para permitir o processo de manipulação de pedidos durante a pesquisa, foi utilizado um computador, com a seguinte especificação: Sistema Operacional Windows 11, processador Intel Core i5, 8GB de memória RAM, disco SSD de 240GB e uma impressora. Adicionalmente, o pesquisador utilizou uma planilha eletrônica especialmente desenvolvida para este estudo, na qual registrava o tempo entre a entrega da lista de medicamentos a ser separada e o retorno do trabalhador com os medicamentos devidamente separados.

4.2.3. Procedimento de coleta de dados

Uma vez que esta pesquisa envolveria trabalhadores humanos, mediante autorização da empresa, foi realizada uma reunião com eles antes do início da coleta. Nesse encontro, foram apresentados os objetivos da pesquisa. Os participantes que concordaram em fazer parte do experimento, assinaram o Termo de Consentimento Livre e Esclarecido, permitindo que os dados fossem coletados. Esta pesquisa foi aprovada por Comitê de Ética em Pesquisa com Seres Humanos, CAAE n. 77411922.3.0000.5302.

O computador do pesquisador gerou randomicamente, a cada solicitação, uma lista com seis medicamentos diferentes que deveriam ser localizados e separados pelos participantes. Essa lista foi impressa e entregue nas mãos de cada participante e o sistema de registro de tempo foi acionado, devendo o separador ler a lista e buscar os remédios.

Na condição Sem Algoritmo (SA), o próprio separador definia a rota que julgasse como a mais adequada. Na condição Com Algoritmo (CA), na folha impressa com a lista de remédios, o separador tinha as instruções da rota que deveria obedecer. Os dados em cada condição foram coletados com um separador por vez. Não alteramos a posição dos medicamentos entre as tarefas de OP ou em relação à sua localização habitual. Assim, os separadores, por serem experientes, tinham a possibilidade de já conhecer de memória a posição do medicamento, o que favorecia a condição SA.

Além disso, não controlamos a distância a ser percorrida em cada tarefa de OP. Na condição SA, a distância dependia da rota definida pelo separador, sendo que ele era livre para estabelecê-la. Na condição CA, não padronizamos as distâncias para cada lista gerada. Contudo, como era feito um sorteio dentre os medicamentos cadastrados, não há motivos para acreditar que as distâncias mínimas possíveis de serem percorridas e o grau de dificuldade da tarefa de OP, entre condições SA e CA, fossem diferentes. A Tabela 2 exibe uma síntese do delineamento proposto.

Tabela 2. Síntese do delineamento de linha de base múltipla seguido por uma reversão.

P	Condição em vigor para a tarefa de <i>order picking</i>							
P1	SA	SA	CA	CA	CA	CA	SA	CA
P2	SA	SA	SA	SA	CA	CA	SA	CA

A coleta foi iniciada com o P1, que foi submetido a duas tarefas de OP na condição SA. Após a primeira etapa, o pesquisador iniciou para o P1 a condição CA, na qual ele realizou mais quatro tarefas de OP. Na sequência, foi realizado o procedimento de reversão, com a retirada da VI, ou seja, P1 voltou para a condição SA por mais uma tarefa



e, em seguida, executou mais uma tarefa na condição CA. Foram aplicadas para P1 oito tarefas de OP. Ao final de cada tarefa de OP para P1, foi realizada a coleta com P2. Esse participante realizou quatro tarefas de OP na condição SA e duas na condição CA, seguida por uma tarefa de reversão na condição SA e uma última tarefa de OP na condição CA.

4.2.4. Procedimento de análise de dados

A análise dos dados consistiu, além da inspeção visual do gráfico, no cálculo da média de tempo e do desvio-padrão em cada condição. A identificação de um tempo menor na condição CA, comparativamente à condição SA, foi adotada como evidência para a hipótese de que a rota de OP, proposta por um algoritmo, supera aquela executada pelo separador.

5. Resultados e Discussão

A Figura 4 exibe dois gráficos de linha. A figura superior é relativa aos dados de P1, e a inferior apresenta os dados de P2. As figuras estão sobrepostas para facilitar a comparação entre os dados dos participantes. Nessa figura, as linhas vermelhas tracejadas indicam o início da condição de *order picking* orientado por algoritmo, enquanto as linhas azuis tracejadas indicam o retorno à condição original de *order picking* sem a orientação de um algoritmo.

Podemos notar na Figura 4, especialmente no caso de P1, uma tendência de redução do tempo de *order picking* na condição de busca orientada por algoritmo. Esse padrão é menos evidente no caso de P2, em parte porque foram coletados poucos dados desse participante na condição sob efeito do algoritmo. Ainda assim, os tempos de *order picking* sob efeito do algoritmo foram menores do que quando a busca foi livre de restrições sobre como percorrer os corredores. Isso fica evidente quando olhamos para as médias de cada participante nas condições com e sem algoritmo. Na condição sem algoritmo, P1 apresentou um tempo médio de *order picking* de 4,49 minutos ($DP = 1,26$), o qual foi reduzido para 2,32 minutos ($DP = 1,38$) na condição com algoritmo. Por sua vez, P2 apresentou média de 4,77 minutos ($DP = 1,09$) na condição sem algoritmo e 3,14 minutos ($DP = 0,59$) na condição com algoritmo. A diferença de tempo para P1 entre as condições foi de 2,17 minutos e para P2 foi de 1,63 minutos.

Com efeito, Elbert et al. (2017) verificaram que quando o separador utiliza de sua experiência para se desviar da rota preestabelecida pelo algoritmo, há aumento da variável tempo. Neste trabalho, pode-se observar que quando P1 segue o algoritmo, há a tendência de o tempo de separação diminuir, e quando usa a própria experiência há tendência de o tempo aumentar. Para P2, esse padrão é menos evidente, mas no geral o seu tempo foi menor na condição com algoritmo.

Urzúa et al. (2019), por sua vez, verificaram redução de 55% no tempo médio de separação, quando humanos eram auxiliados pela máquina. No resultado deste estudo, nota-se que P1, teve, em média, uma melhora de 48,33% no tempo de separação, enquanto P2 obteve melhora de 34,17%. Comparativamente, podemos perceber uma aproximação dos resultados deste estudo com o estudo de Urzúa et al. (2019), que também apresentou redução de tempo de OP na condição com auxílio do algoritmo. Assim, os dados do presente estudo corroboram achados da literatura.

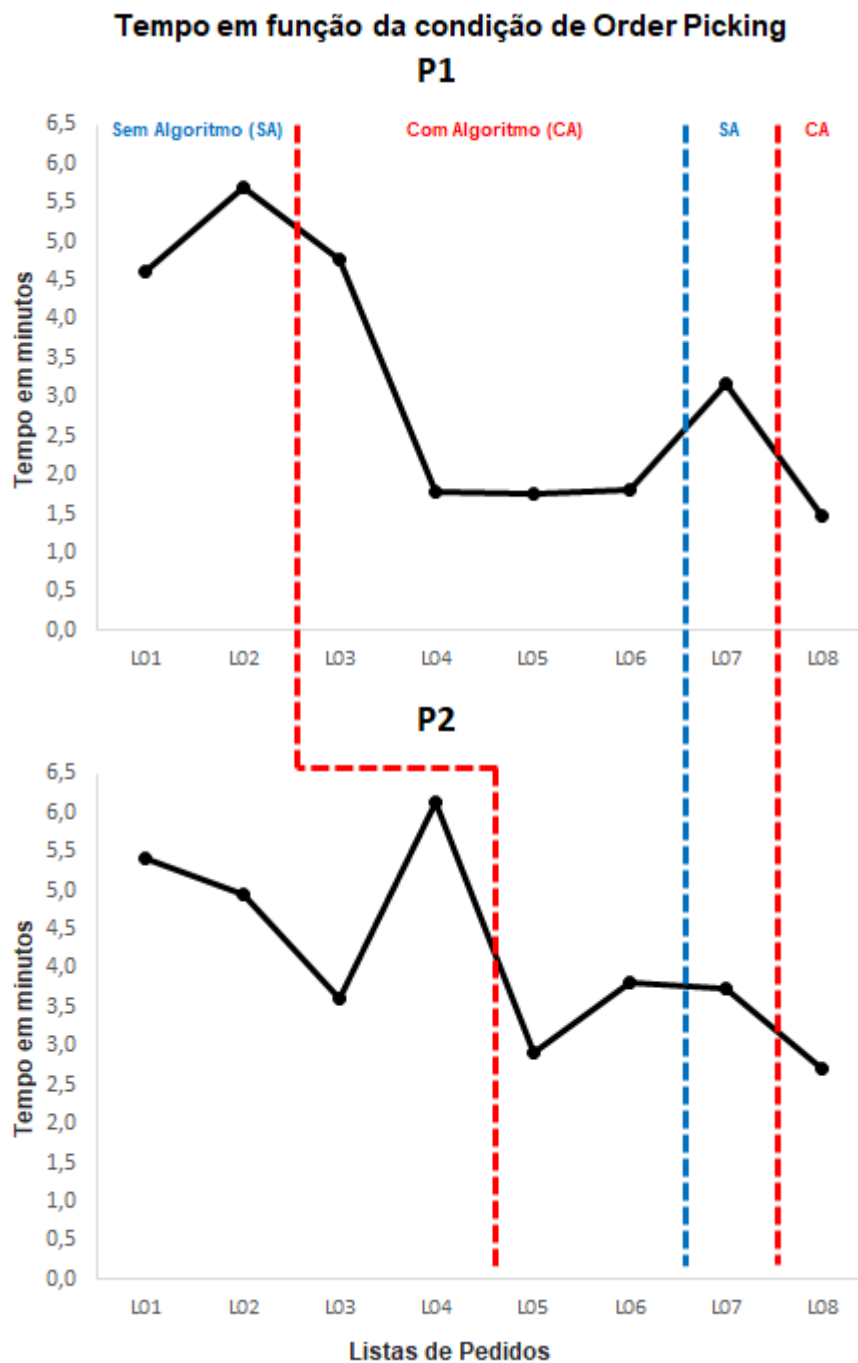


Figura 4. Tempo em minutos em função das condições de *order picking* com e sem auxílio de algoritmo.

Ozden et al. (2020), por sua vez, mostraram que humanos podem ter políticas de rota mais racionais do que o tradicionalmente previsto por modelos de OP, uma vez que “cortam os cantos”. Neste estudo, porém, identificamos uma política de rotas menos eficiente, relacionada a repetir posições já visitadas. Para além disso, não foi possível realizar uma comparação direta com Ozden et al. (2020), visto que não avaliamos a variável distância percorrida neste estudo.

Cumpramos destacar que, durante a condução desta pesquisa em cenário real, com todas as variáveis ambientais influenciando a tarefa de OP, foi possível observar variação



no tempo de separação, sendo que, em apenas um caso, o tempo do algoritmo ficou maior que o tempo sem algoritmo. Tal variação pode estar relacionada a variáveis em vigor no contexto da pesquisa: (1) alta demanda de separação dentro da área de *picking*; e (2) outros trabalhadores não participantes da pesquisa locomovendo-se pelos mesmos espaços que os separadores participantes da pesquisa, o que causou congestionamento nos corredores. Além disso, cada participante realizou oito listas de separação na sequência, o que pode ter sido cansativo. Todos esses fatores podem influenciar o padrão de deslocamento do separador e até mesmo a sua atenção, mesmo que esteja seguindo uma ordem de separação estabelecida por algoritmo.

Esses fatores de distração, foram identificados em pesquisas como a de Ozden et al. (2020), que estabelece que o comportamento do separador ao não seguir a rota preestabelecida produz aumento do tempo de separação. Se o tempo de separação diminui quando o separador “corta um canto”, então, provavelmente, o tempo aumenta quando o separador faz um caminho mais longo ou precisa se desviar de obstáculos no meio do caminho.

Apesar das limitações de controle experimental de todas as variáveis, entendemos que os dados sugerem que o *order picking*, auxiliado pelo algoritmo que implementa a heurística *S-Shape*, e o endereçamento do armazém de modo a orientar a atividade de *picking*, tendem a produzir aumento da eficiência de separação. O grau em que essas duas variáveis impactam nessa atividade, ainda não está claro, devendo ser objeto de estudos futuros. Acreditamos, no momento, que no caso deste estudo, o endereçamento foi uma variável preponderante, uma vez que eliminou os erros do separador ao entrar em corredores inadequados ou a necessidade de olhar para todas as prateleiras e todos os medicamentos.

Por fim, foram encontradas algumas limitações durante a pesquisa que podem ser corrigidas ou atenuadas para estudos posteriores, a saber: (1) A falta de endereçamento prévio para guiar os separadores, quando executando a operação de *order picking* sem auxílio do algoritmo. Nesse caso, a condição do estudo com efeito do algoritmo, tende a ter vantagem em termos de eficiência, pois fica explícita para o separador a posição exata do item a ser coletado; (2) Dispor de mais tempo para a coleta de dados, pois com mais dados, seria melhor a compreensão das variáveis pesquisadas; (3) Contabilizar erros e acertos de coleta, pois na versão atual é possível somente medir a variável tempo, no entanto, não são mensurados possíveis erros de coleta de itens; (4) Aumentar o número de participantes; e (5) Diminuir a quantidade de distrações e desvios no processo de *order picking*, mantendo os separadores concentrados somente na tarefa que estão executando.

Apesar da limitação relativa ao número de participantes, lembramos que o delineamento de linha de base múltipla é um tipo de desenho experimental adequado para esse tipo de situação, especialmente, quando precisamos entender de que modo uma variável independente (VI) impacta uma variável dependente (VD) para uma pessoa específica (Sampaio et al., 2008). Por ser um delineamento de sujeito único, o controle do experimento não depende de um grupo, mas de medidas feitas acerca do comportamento de cada indivíduo antes de uma intervenção. Além disso, ele lida melhor com as variáveis que não conseguimos controlar e que podem afetar o comportamento. Isso ocorre porque realizamos várias medidas da VD na presença e na ausência da VI, superando os delineamentos com apenas duas medidas, pré e pós-teste, por exemplo.



Quando medimos várias vezes, aumentamos as chances de detectar um padrão no comportamento, mesmo em face de alguma variabilidade.

Adicionalmente, ao comparar os dados dos participantes, vemos que a mudança mais expressiva sobre o comportamento ocorre apenas a partir da inserção da VI. Contudo, conforme vimos no caso de P2, a redução no número de medidas da VD pode dificultar a interpretação dos dados, motivo pelo qual é importante dispor de tempo e equipe para ampliar um pouco mais a coleta até que se atinja, em cada condição de presença ou ausência da VI, o que a literatura chama de critério de estabilidade do comportamento. Isso ocorre quando, apesar da variabilidade comportamental, conseguimos identificar um padrão.

6. Conclusão

Neste trabalho, buscou-se investigar a eficiência da atividade humana na tarefa de *order picking* com auxílio de algoritmo de determinação de melhor rota de busca. Foi implementado em JavaScript o algoritmo baseado na heurística *S-Shape*, para determinar a sequência que os itens de uma ordem de separação deveriam ser recolhidos. Essa condição foi confrontada com a busca de produtos sem auxílio de nenhum algoritmo preestabelecido, em que o separador buscava os itens na ordem que ele julgava ser mais eficiente. Ao utilizar o algoritmo, introduziu-se também o endereçamento no estoque, indicando precisamente onde o produto estava localizado dentro do armazém. Comparando a eficiência entre as duas condições, verificamos que: (1) o separador realiza a busca de produtos a partir do que se recorda, o que pode ser ineficiente; e (2) quando o separador realiza a busca com auxílio do algoritmo que implementa a heurística *S-Shape*, os dados sugerem que há aumento da eficiência de separação.

Por fim, sugere-se como trabalhos futuros a replicação desta pesquisa, com endereçamento definido, comparando, dessa vez, a eficiência entre dois algoritmos pré-estabelecidos, analisando o comportamento humano ao seguir ou não o estabelecido por essas políticas de rota. Para esses estudos, sugere-se aumentar os números de separadores participantes, a quantidade de listas e registrar tempo e distância como variáveis dependentes. Esperamos, assim, por meio deste trabalho contribuir para o aumento da eficiência dos processos de OP de centros de distribuição de pequeno e médio porte.

7. Referências

- Abdulkareem, S. A., & Abboud, A. J. (2021). Evaluation Python, C++, JavaScript and Java programming languages based on software complexity calculator (halstead metrics). *IOP Conference Series: Materials Science and Engineering*, 1076. <https://doi.org/10.1088/1757-899X/1076/1/012046>
- Cano, J. A., Correa-Espinal, A. A., & Gómez-Montoya, R. A. (2017). An evaluation of picking routing policies to improve warehouse efficiency. *International Journal of Industrial Engineering and Management*, 8(4), 229-238. <https://doi.org/10.24867/IJEM-2017-4-123>
- Dijkstra, A. S., & Roodbergen, K. J. (2017). Exact route-length formulas and a storage location assignment heuristic for picker-to-parts warehouse. *Transportation Research Part E: Logistics and Transportation Review*, 102, 38-59. <https://doi.org/10.1016/j.tre.2017.04.003>



- Elbert, R. M., Franzke, T., Glock, C., & Grosse, E. H. (2017). The effects of human behavior on the efficiency of routing policies in order picking: The case of route deviations. *Computers & Industrial Engineering*, *111*, 537–551. <https://doi.org/10.1016/j.cie.2016.11.033>
- Grosse, E. H., Glock, C. H., & Neumann, W. P. (2016). Human factors in order picking: a content analysis of the literature. *International Journal of Production Research*, *55*, 1260-1276. <https://doi.org/10.1080/00207543.2016.1186296>
- Hall, J. (2022). Getting Started with MongoDB & Mongoose. MongoDB. Disponível em: <https://www.mongodb.com/developer/languages/javascript/getting-started-with-mongodb-and-mongoose/#what-is-mongoose>
- Masae, M., Glock, C., & Grosse, E. H. (2019). Order picking routing in warehouses: A systematic literature review. *International Journal of Production Economics*, *224*, 107564. <https://doi.org/10.1016/j.ijpe.2019.107564>
- Microsoft. (2022). Visual Studio Code, getting started. Visual Studio Docs. Disponível em: <https://code.visualstudio.com/docs>
- MongoDB. (2022). What is MongoDB? MongoDB Documentation. Disponível em: <https://www.mongodb.com/docs/manual/>
- Mozilla. (2022). Sobre JavaScript. MDN Web Docs. Disponível em: https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/About_JavaScript
- OpenJS Foundation. (2022). Introduction to Node.js. Nodejs.dev. Disponível em: <https://nodejs.dev/en/learn/>
- OpenJS Foundation. (2022). Node.js. Disponível em: <http://nodejs.org>
- Ozden, S. G., Smith, A. E., & Gue, K. R. (2021). A novel approach for modeling order picking paths. *Naval Research Logistics*, *68*, 471-484. <https://doi.org/10.1002/nav.21966>
- Pano, A., Graziotin, D., & Abrahamsson. (2018). Factors and actors leading to the adoption of a JavaScript framework. *Empirical Software Engineering*, *23*, 3503-3534. <https://doi.org/10.1007/s10664-018-9613-x>
- Redmer, A. (2020). Analysis of the length of order-picking paths determined using the s-shape method. *Logforum Scientific Journal of Logistics*, *16*(1), 33-46. <http://doi.org/10.17270/J.LOG.2020.379>
- Sampaio, A. A. S., Azevedo, F. H. B., Cardoso, L. R. D., Lima, C., Pereira, M. B. R., & Andery, M. A. P. A. (2008). Uma introdução aos delineamentos experimentais de sujeito único. *Interação em Psicologia*, *12*(1), 151-164. <http://dx.doi.org/10.5380/psi.v12i1.9537>
- Sun, K., & Ryu, S. (2018). Analysis of JavaScript Programs: Challenges and Research Trends. *ACM Computing Surveys*, *50*(4), 1-34. <https://doi.org/10.1145/3106741>
- Urzúa, M., Mendoza, A., & González, A. O. (2019). Evaluation the impact of order picking strategies on the order fulfillment time: A simulation study. *International Scientific Journal about Logistics*, *6*, 103-114. <https://doi.org/10.22306/al.v6i4.129>



- Valle, C. A., Beasley, J. E., & Da Cunha, A. S. (2017). Optimally solving the joint order batching and picker routing problem. *European Journal of Operational Research*, 263(3), 817-834. <https://doi.org/10.1016/j.ejor.2017.03.069>
- Van Gils, T., Ramaekers, K., Braekers, K., Depaire, B., & Caris, A. (2018a). Increasing order picking efficiency by integrating storage batching, zone picking, and routing policy decisions. *International Journal of Production Economics*, 197, 243-261. <https://doi.org/10.1016/j.ijpe.2017.11.021>
- Van Gils, T., Ramaekers, K., Caris, A., & Koster, R. B. M. (2018b). Designing efficient order picking systems by combining planning problems: State-of-the-art classification and review. *European Journal of Operational Research*, 267(1), 1-15. <http://dx.doi.org/10.1016/j.ejor.2017.09.002>
- Winkelhaus, S., Grosse, E. H., & Morana, S. (2021). Towards a conceptualization of Order Picking 4.0. *Computers & Industrial Engineering*, 159. <https://doi.org/10.1016/j.cie.2021>