IEEE Access
Multidisciplinary Rapid Review Open Access Journal

Received 10 September 2024, accepted 27 October 2024, date of publication 13 November 2024, date of current version 26 November 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3497592

RESEARCH ARTICLE

Optimising Warehouse Order Picking: Real Case Application in the Shoe Manufacturing Industry

RODRIGO FURLAN DE ASSIS [1], **WILLIAM DE PAULA FERREIRA** [1], **ALEXANDRE FRIAS FARIA** [1], **LUIS ANTONIO SANTA-EULALIA** [2], **MUSTAPHA OUHIMMOU** [1], **AND ALI GHARBI** [1]

[1]Department of Systems Engineering, École de Technologie Supérieure, Montreal, QC H3C 1K3, Canada
[2]Business School, Université de Sherbrooke, Sherbrooke, QC J1K 2R1, Canada

Corresponding author: William de Paula Ferreira (william.ferreira@etsmtl.ca)

This work was supported in part by Mitacs through the Mitacs Accelerate Program in Canada, and in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada under Grant RGPIN-2024-03880.

ABSTRACT Order picking is a critical and labour-intensive warehouse management operation that involves removing items from storage locations to fulfil customer orders. This paper analyses a new order-picking problem based on the real case of a Canadian shoe manufacturer characterised by a warehouse with random storage, where different product types can be assigned to a single storage location. While maximising space utilisation, considering the high number of Stock Keeping Units, this storage approach makes the creation of efficient picking routes challenging, increasing the effort needed to complete picking orders. To address this challenge, we present the Genetic Route Optimisation algorithm for optimising order-picking routes. Our methodology involved testing the proposed algorithm using real-world data derived from the company's Warehouse Management System. The results demonstrate a reduction in picking distances, highlighting the effectiveness of the Genetic Route Optimisation algorithm in optimising picking routes in a random storage environment. As well as presenting a practical application case, the study highlights the potential of the proposed algorithm to improve operational efficiency in warehouse environments. It also paves the way for future research in warehouse logistics, especially by adapting similar algorithmic strategies to various complex and dynamic warehouse environments, thus advancing the field of warehouse management.

INDEX TERMS Random storage location, mixed shelving, picker routing, genetic algorithm.

I. INTRODUCTION

To meet customer demand and stand out, warehouses face various challenges when developing their operational strategy [1], [2]. In supply chains where product handling is critical, warehouse activities such as receiving, storage, order picking (OP) and shipping are fundamental for operational efficiency [3]. Among these, OP - commonly defined as removing items from their storage locations in response to customer orders - is considered one of the most time-consuming and labour-intensive [4]. In most cases, a customer order is transformed into a picking list, which includes information on the location, quantity, and order in which items are to be picked, a process known as discrete order picking [5]. In particular, picker-to-part OP systems are significant resource users, where operators travel to product locations to pick them up manually [6]. This type of system is widely adopted for its flexibility and capability to manage diverse items and orders while requiring a lower investment [7]. However, the efficiency of these systems depends heavily on optimising the picking routes [8].

The literature on OP identifies four main sub-challenges: order batching, batch assignment, batch sequencing, and picker routing [9]. Concerning picker routing - which is the focus of this study - the main objective is to minimise the total distance travelled due to its high proportion in total picking time. However, other performance indicators are equally important [3]. These include reducing delays [10], reducing fatigue [11], minimising queues [9], using resources efficiently [12], increasing picking productivity [13], and

The associate editor coordinating the review of this manuscript and approving it for publication was Frederico Guimarães.

optimising work in progress [14]. However, distance travelled remains one of the most critical indicators, as it is directly related to travel time and overall process efficiency, especially with arbitrary warehouse configurations, including multiple high storage areas and multiple parallel aisle warehouses [2].

This emphasis on reducing travel distance highlights the complexity of warehouse layout and product location [15]. Specifically, product location's arbitrary and often unpredictable nature, especially in warehouses with randomised storage systems, presents significant challenges [16]. Optimising picking routes becomes a challenging task in a random storage environment, where different types of products can be assigned to a single storage location [17]. Operators must travel longer distances to pick up the required items, increasing the time and effort needed to complete orders [18]. In addition, the lack of a fixed standard for product location makes creating efficient and consistent routes difficult, requiring dynamic and adaptable approaches to minimise travel distances [19], [20].

The logistics industry has a growing interest in the random storage strategy, since it significantly optimises warehouse space, allowing companies to accommodate more products without the need for physical expansion [21]. In addition, it offers unprecedented flexibility in inventory management, making it easier to adapt to changes in market demand [22]. However, while these managerial and operational advantages are widely documented, they introduce substantial complexities in picking routing [23]. The unpredictable nature of product location and the need for adaptable picking routes make route optimisation a considerable challenge [24].

A particularly challenging aspect of random storage arises when a single storage position can contain multiple products. This scenario amplifies the complexity of the order-picking process by requiring more sophisticated route optimisation strategies to handle the increased variability and unpredictability in product locations. While there is existing research on order picking in environments with mixed storage [25], our study addresses a specific operational context where combining random storage and multiple products per position necessitates new approaches. This distinction characterises a new OP problem in the literature because it integrates the challenges of randomised storage location with the need to optimise routes for multiple products within single storage locations, which needs to be sufficiently explored in practical warehousing scenarios [26].

This gap underscores the relevance of our paper, inspired by a real industrial case observed by the authors in a Canadian shoe manufacturer warehouse with a mixed shelving system, where each storage position can contain multiple products. Based on the characteristics of the warehouse, this paper proposes the Genetic Route Optimisation Algorithm (GRO). In addition, we provide a practical framework for implementing picking routing strategies, detailing the steps required to adapt and apply the GRO algorithm in real operational scenarios. To achieve this, we analysed the process developed and its application in the specific operational context, in which we determined a route to reduce travel distances in the warehouse and establish a picking order to facilitate movement in the warehouse by simplifying the sequence in which items are selected for orders. A key contribution of our research is the proposition of using GRO to solve a real-life problem using real data and situations encountered in a real warehouse environment. This approach offers practical insights and solutions that can be directly applied in similar industrial contexts. By working with real-world instances and data, our study bridges the gap between theoretical research and practical application, providing a tested solution for optimising order-picking processes in warehouses with random storage systems and multiple products per storage position.

In the following sections, we describe the picking route problem. Section III reviews the related literature. Section IV presents the GRO framework we designed. Section V applies our approach to the warehouse configuration of a shoe manufacturing company using a dataset to evaluate its effectiveness in generating improved solutions to the problem. Section VI discusses GRO performance. To conclude, we summarise the study's main findings, discuss its implications and limitations, and offer suggestions for future research in Section VII.

II. PROBLEM DESCRIPTION AND FORMULATION

Our study aimed to increase the efficiency of the picking route process in a warehouse characterised by a complex layout and random storage policy. This warehouse features multiple blocks, parallel aisles, and multi-level racking systems, all stocking a diverse range of products. The random storage location assignment policy adds to the complexity of the task, as each item may be located in different storage locations, and each storage location may contain multiple products. This configuration poses a challenge to devising an effective route planning strategy, as it requires navigating through a constantly changing set of locations. Specifically, we aimed to minimise the travel distance required for order picking. By reducing the distance travelled, we sought to enhance the performance of the order-picking process. Additionally, we aimed to generate a visual solution to facilitate the identification and navigation of optimal picking routes within the warehouse. It is intended to assist operators in following the most efficient paths, thereby streamlining the order-picking process and reducing operational time and effort.

Figure 1 shows the parallel-aisle warehouse studied in this article. It contains cross aisles to the left and right of the picking aisles, as in Figure 1(a), and includes a central cross aisle that divides the warehouse perpendicularly and, therefore, divides the aisles into picking sub-aisles. The warehouse consists of three blocks: A, B and C. Block A has seven sections, each containing six shelves with double-sided storage spaces, a configuration repeated in block B. However, block B has nine sections, as does block C. Each section can accommodate a variety of products. The aisles in the warehouse are organised in vertical and horizontal lines that meet at specific points. These intersection points are indicated in blue circles, forming the pathways of the picking aisles and facilitating efficient movement and item retrieval in the warehouse. These aisles contain sets of ''item points'', which determine the specific positions for picking items from each section's shelves.

Figure 1(b) shows a representation of the warehouse under analysis, providing a visual perspective to understand the spatial layout and structural characteristics relevant to the study. This system directly retrieves items from shelves comprised of four levels. Levels three and four primarily house stocks to replenish levels one and two, where picking items are stored. The nature of the work requires orders to be collected manually. In particular, products are distributed randomly on the racks, accessible from both sides, maximising the use of space but presenting a unique challenge for the picker-to-part order-picking system. In addition, the warehouse has a single entry/exit (I/O) point, which serves as the start and end point for all pickers. Order picking is carried out in waves, with each picker responsible for a specific set of assigned orders. The pickers have a trolley, capable of carrying up to 25 items, to help organise and transport the collected items to the drop-off point. An Automatic Guided Vehicle (AGV) takes the loaded trolleys to the expedition area. This method is strategic for managing workflow and optimising picking time. However, the efficiency of this approach depends on the route chosen by each picker to complete their task.

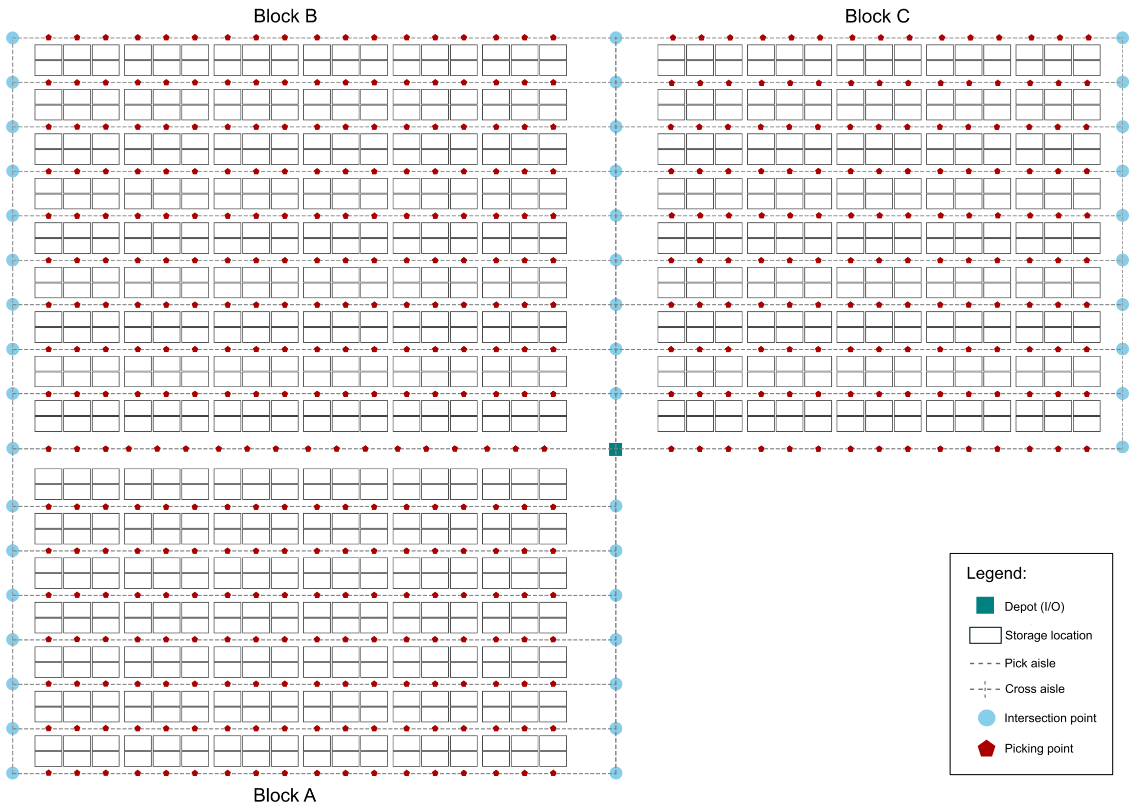
Additionally, we use a representation that allows each position (i) to be designated by its coordinates (x_i, y_i, z_i), signifying its position along the x and y axes and its z-level in the storage system. Within these storage positions, products are inserted randomly and can vary in quantity, with each position accommodating up to eighteen distinct products (i.e., boxes containing pairs of boots). This randomness in product placement within positions adds complexity to the optimisation of picking routes, requiring algorithms to adapt to variable item retrieval sequences.

Although routing heuristics are a widely adopted approach to optimising picking routes in warehouses, as discussed by [27], we chose not to use them in this study, prioritising the minimisation of distance travelled. This decision is based on the specific nature of our target warehouse, which has a complex multi-block layout and a randomised stock system with multiple products on the shelves. In such scenarios, the effectiveness of conventional routing heuristics is often limited due to the unpredictability and variability of item location [28]. Nevertheless, minimising the travel distance offers an objective and quantifiable criterion that can be optimised more effectively in a non-deterministic storage environment [29]. By reducing the distance travelled by the pickers, we aim to develop a solution that improves operational efficiency to deal with the complexity and randomness inherent to the warehouse configuration with multiple products on each storage position. This strategy allows us to formulate a solution approach directly aligned with the specific characteristics and operational challenges of the warehouse studied, focusing on practical applicability in storage environments.

Our warehouse logistic operations face a routing challenge extending beyond the traditional Travelling Salesman Problem (TSP) or its variant, the Family Travelling Salesman Problem (FTSP). The FTSP is a complex variant of the well-known TSP, which seeks to identify the most efficient route to visit a set of locations. However, in the FTSP, these locations are grouped into distinct sets or 'families,' which requires visiting each family of locations in a specific sequence [30]. The key distinction from the TSP lies in the FTSP requirement to not only visit each location within a family at least once but also to ensure that all members of a given family are visited in succession before moving on to the next family and ultimately returning to the starting point [31]. This sequential visiting of families adds a layer of complexity to route optimisation, accommodating the unique operational challenges presented by our warehouse [32].

In the picking operations within our warehouse, we customise the FTSP to better suit the picking process's specifics. Here, ''families'' are defined as locations where specific products are stored, as listed for picking. Unlike the standard FTSP, where each family is a separate group with no overlap, our warehouse scenario introduces a complexity where the same product might be found in multiple locations, leading to potential overlaps between these families. Consequently, the challenge is to traverse these overlapping families to pick all listed items efficiently. To effectively address the challenges posed by random storage and multiple products in a single position, we adapted the FTSP to reflect the specificities of our picking process. This adaptation considers the reality of our warehouse, where individual storage bins may house various items, often stored on multi-level shelves and accessible from both sides of the aisle. By integrating this feature into the FTSP, we introduce a new layer of complexity: the need to navigate efficiently through locations with groupings of several products. The main objective of this strategic adaptation is to optimise the picking route, simplifying route planning to minimise distances travelled while ensuring adequate access to these densely populated storage areas.

Thus, our problem is formulated as follows: given a list of products to be picked in a warehouse with random storage and multiple products per position, what is the optimal sequence for visiting the locations to minimise the total length of the picker's route? The picker routing challenge is a complex optimisation problem with direct applications in warehouse management and order picking systems [33]. Specifically, this problem involves solving two interconnected optimisation sub-problems: first, determining the optimal order in which to visit the pick points, taking into account the overlap of product ''families'', and second, establishing the most efficient route for the picker to travel



(a) Warehouse layout showing the storage locations (1st and 2nd level) and picking positions.

(b) Warehouse photo.

FIGURE 1. **Schematic layout of the warehouse (a) and photo of the warehouse (b) with multiple products per storage location.**

between these selected points, ensuring adequate access to areas with clusters of various products [32]. This formulation accounts for our warehouse's unique storage setup and the practical need to optimize picking routes.

The optimisation problem is formulated as follows, adapted from Daniels et al. [34] and Moran et al. [30]. Let P represent the set of products that need to be picked, also known as the picking list, and V denote the vertices representing specific positions in front of the warehouse shelves. The set E consists of edges representing the aisles or cross-aisles that structure the layout of the warehouse, allowing us to construct a graph $G(V, E)$ that represents this layout comprehensively. For each product p, S_p defines the set of all positions where that product can be found. Additionally, we define $D = [d_{i,j}]$ as the adjacency matrix associated with the graph $G(V, E)$, where each entry $d_{i,j}$ corresponds to the distance between positions i and j within the warehouse. The quantity of product p available at position i is denoted by $q_{p,i}$, while r_p represents the quantity of product p required according to the picking list P. The optimisation problem is formulated as follows:

$$\min \sum_{(i,j)\in E} d_{i,j} z_{i,j} \tag{1}$$

Subject to the following constraints:

$$\sum_{i\in S_p} q_{p,i} y_i \geq r_p, \quad \forall p \in P \tag{2}$$

$$\sum_{i\in V\setminus\{j\}} z_{i,j} = y_j, \quad \forall j \in V \tag{3}$$

$$\sum_{j\in V\setminus\{i\}} z_{i,j} = y_i, \quad \forall i \in V \tag{4}$$

$$\sum_{i\in V(C)} \sum_{j\in V(C)} z_{i,j} \leq |C| - 1, \quad \forall C \in \mathcal{C}^t \tag{5}$$

$$z_{i,j} \in \{0, 1\}, \quad \forall (i, j) \in E \tag{6}$$

$$y_i \in \{0, 1\}, \quad \forall i \in V \tag{7}$$

The objective function in equation (1) minimises the total travel distance by the picker. Here, $d_{i,j}$ denotes the distance between two positions i and j in the warehouse, and $z_{i,j}$ is a binary variable set to 1 if the picker travels directly from position i to j. Thus, this objective function sums up the distances for each step in the picker's route, aiming to minimise the total travel distance. Equation (2) represents the demand fulfilment constraint, ensuring that a sufficient number of positions are selected to cover the required quantity r_p for each product p in the picking list P. This is achieved by summing the quantities $q_{p,i}$ available at the selected positions i (where $y_i = 1$) and ensuring that this sum meets or exceeds r_p. Equations (3) and (4) are the flow continuity constraints, which maintain route continuity. Equation (3) ensures that if a position j is selected for the route (i.e., $y_j = 1$), exactly one edge $z_{i,j}$ must lead into j. Similarly, equation (4) requires exactly one edge $z_{i,j}$ to lead out of each selected position i. Together, these constraints guarantee that each selected position is entered and exited once, ensuring a continuous route. Equation (5) enforces sub-tour elimination by preventing disconnected cycles within the picker's route. Here, $V(C)$ denotes a subset of nodes, and this constraint ensures that any subset C cannot form a closed loop unless it includes all nodes in the full route. This guarantees a single, continuous tour that spans all selected positions. Finally, equations (6) and (7) define the binary nature of the decision variables. Specifically, $z_{i,j}$ takes a value of 1 if the picker directly travels from position i to j, and 0 otherwise. Similarly, y_i is set to 1 if position i is included in the picker's route and 0 otherwise. These binary constraints clarify which positions and connections are part of the final optimized route. Despite the model's effectiveness in producing optimal picking routes, the computation time required to solve them is high. In particular, solution times have increased significantly for instances involving more than four families, which is in line with the results of Moran et al. [30].

III. LITERATURE REVIEW

Solution approaches for picker routing can be differentiated into exact, heuristic, metaheuristic algorithms, and simulation [35], [36], [37], [38]. Although exact algorithms provide a straightforward solution to the problem, it is essential to recognise that they generally face scalability and computation time challenges, especially when dealing with large-scale instances of the problem [39]. Heuristic routing policies have been instrumental in tackling the challenges of routing order pickers in warehouses, but their inherent limitations in guaranteeing optimal solutions have led researchers to explore advanced methodologies, emphasising metaheuristic approaches that offer ways to improve solutions [40]. By taking advantage of iterative processes and heuristic-driven strategies, metaheuristics aim to overcome local optima by considering larger solution spaces to refine routing solutions [41]. In addition, the complexity inherent in the combinatorial nature of this problem makes the application of metaheuristics a viable alternative [42]. These methods offer adaptive and exploratory approaches to optimise routes in challenging environments, such as those with a random arrangement of products [43].

The use of metaheuristics in the picking route problem has been widely explored in the literature, demonstrating the effectiveness of these approaches in various storage scenarios. Among the most widely used metaheuristics are genetic algorithms (GAs) [44], variable neighbourhood search (VNS) [45], particle swarm solution (PSO) [46], ant colony optimisation (ACO) [47], and hybrids approaches [48]. In conducting this study, we adopted a systematic review approach based on the methodology proposed by Massae et al. [2], which provides a comprehensive framework for identifying relevant research on exact algorithms, heuristics and metaheuristics in the context of optimising picking routes in warehouses. With a specific focus on metaheuristics, we conducted a thorough search of the relevant literature published up to 2024. We reworked the proposed protocol, selecting studies that use these techniques to solve picking routing problems. This process resulted in the selection in Table 1, representing a broad spectrum of metaheuristic approaches and applications. The Table 1 categorises the selected studies based on critical characteristics, such as warehouse configuration, storage system, level of mechanisation, specific objectives, and methodologies applied, as proposed by [49].

Each category in Table 1 represents an essential attribute of the studies, providing a detailed context for each approach. "Warehouse" refers to the structural layout, such as single-block or multi-block designs, which impacts routing complexity. The "Storage" indicates whether products are stored in dedicated or random locations, influencing retrieval strategies. The "Mechanisation" differentiates between manual, semi-automated, and fully automated environments, affecting the choice of optimisation techniques. "Objective" captures the primary goals—typically minimising travel distances, picking time, or operational costs. Finally, "Method"

TABLE 1. Related studies.

Reference	Warehouse				Storage				Mechanisation			Objective (Minimise)	Method
	1	2	3	4	5	6	7	8	9	10	11		
Tsai et al. [44]	✓		✓		✓				✓			Operational cost	GA
Rubrico et al.[50]		✓	✓		✓						✓	Makespan	Tabu search
Chen et al. [51]		✓	✓		✓						✓	Picking time	ACO
Chen et al. [47]	✓		✓		✓						✓	Picking time	ACO & GA
Lin et al. [46]	✓		✓		✓						✓	Picking time	PSO
Cortés et al. [52]	✓			✓	✓				✓			Picking time	Generic tabu search
Schrotenboer et al. [53]		✓		✓	✓				✓			Travelling distance	Hybrid GA
Chabot et al. [54]		✓	✓		✓				✓			Travelling distance	Large neighbourhood search
Li et al. [55]	✓		✓		✓						✓	Travelling distance	ACO
Ardjmand et al. [10]		✓	✓		✓				✓			Travelling distance	ACO & GA
De Santis et al. [5]		✓	✓		✓				✓			Travelling distance	ACO
Bódis & Botzheim [56]		✓	✓		✓						✓	Travelling distance	Bacterial Memetic Algorithms
Weidinger [57]	✓		✓				✓		✓			Picking time	Nearest neighbor heuristic
Ardjmand et al. [58]	✓		✓		✓						✓	Travelling distance	Simulated annealing & GA
Chen et al. [48]		✓	✓		✓						✓	Travelling distance	ACO & GA
Van Gils et al. [59]		✓		✓			✓		✓			Travelling distance	Large neighbourhood search
Bottani et al. [60]		✓	✓		✓				✓			Travelling distance	Harmony search
Gil-Borrás et al. [61]		✓	✓			✓	✓		✓			Travelling distance	VNS
Düzgit et al. [62]		✓	✓			✓			✓			Travelling distance	Iterated greedy algorithm
Wu et al. [45]	✓		✓		✓				✓			Travelling distance	Simulated annealing
Cergibozan & Tasan [63]		✓		✓	✓						✓	Travelling distance	GA
Cano et al. [64]		✓		✓		✓			✓			Picking time	ACO & GA
Wu et al. [65]	✓			✓			✓		✓			Travelling distance	GA & local search
This study		✓		✓		✓		✓	✓			Travelling distance	**Adapted GA**

Legend: (1) Single-block; (2) Multi-block; (3) Single-floor; (4) Multi-floor; (5) Dedicated; (6) Random; (7) Classes; (8) Multiple products in a single position; (9) Manual; (10) Automated; (11) Semi-Automated.

lists the specific metaheuristic approaches, highlighting the diversity of algorithms used to address the varied challenges in optimising warehouse picking.

Tsai et al. [44] developed a dual genetic algorithm (GA) approach to address the challenges of both order batching and picker routing, aiming to minimise travel costs and manage early/late penalties effectively. The study first applies a GA to batch orders, followed by a second GA to determine the optimal route for the picker, tailored to the specific items in each batch. Conducted in a warehouse with a rectangular layout and parallel aisles, the research focuses on a dedicated storage system where each location accommodates only one type of item, stored exclusively in a single position. In a similar line of research, Rubrico et al. [50] utilized a tabu search algorithm to address picker routing, prioritizing the reduction of total execution time, or makespan, within a dedicated storage setup. Chen et al. [51] later applied an ant colony optimisation (ACO) technique to minimise customer order delays, building on dedicated storage principles. Extending this work, Chen et al. [47] proposed a hybrid model that combines ACO and GA to reduce picking time and alleviate congestion in warehouses with narrow aisles. Both Chen et al. studies highlight the effectiveness of combining metaheuristic approaches to optimize picker routing in dedicated storage environments.

Chabot et al. [54] investigated large neighbourhood search techniques to reduce travel distances in warehouses with dedicated storage systems, demonstrating the method's effectiveness in optimizing picker routes. Bottani et al. [60] later applied Harmony Search algorithms to similar dedicated storage contexts, further optimising travel distances and overall warehouse efficiency. In a comprehensive study, Cortés et al. [52] tackled the order picker routing problem with an emphasis on creating optimized tours, accounting for multiple factors such as product attributes (e.g., weight and volume), storage location heights, inventory levels, and the diversity of material handling equipment available. Bódis and Botzheim [56] used a bacterial memetic algorithm to optimize picker routing by focusing on pallet load characteristics, item properties, and pick list structures. Their approach emphasized ensuring stable pallet configurations to prevent product damage, using a matrix-based format to illustrate optimal pallet loading and pick sequences.

Li et al. [55] used an ACO approach combined with local search to solve the order routing problem in a warehouse with two blocks and a single storage location. De Santis et al. [5] introduced a new hybrid metaheuristic (combines ACO with the Floyd-Warshall algorithm) approach to improve order picker routing in a narrow-aisle warehouse scenario characterised by two distinct blocks operating in a single low-level storage environment. Chen et al. [48] developed a hybrid algorithm combining ACO with GA to optimise order picking in multi-block warehouses characterised by ultra-narrow aisles and access restrictions. Van Gils et al. [59] used a large-neighbourhood search to address similar challenges in automated multi-block warehouses. Schrotenboer et al. [53] present a hybrid GA designed to optimise picking and return-to-stock routes, considering interactions between order pickers in multi-block warehouses.

Ardjmand et al. [10] addressed logistical challenges in rectangular, single-block warehouses by combining Lagrangian decomposition heuristics with particle swarm optimisation (PSO). This approach tackled order grouping, batch assignment, and order picker routing, focusing on minimising the time required to complete all order batches, considering the involvement of multiple operators. Lin et al. [46] also examined the joint optimisation of order batching and picker routing within a single-block warehouse with a single depot. Further extending this work to multi-block warehouse environments, recent studies, including Wu et al. [65], Wu et al. [45], Cergibozan and Tasan [63], Gil-Borrás et al. [61], and Ardjmand et al. [58], have also combined these optimisation problems, highlighting the complexity and scalability considerations unique to multi-block configurations.

Cano et al. [64] conducted a study focused on minimising travel time within high-level, multi-block storage systems by solving the picker routing problem (PRP) through the use of genetic algorithms (GA) and ant colony optimisation (ACO). This approach considers height restrictions and aisle configurations unique to complex warehouse layouts. Düzgit et al. [62] proposed a hybrid metaheuristic combining tabu search and an iterated greedy algorithm to optimize order-picking in a multi-block warehouse, targeting the most efficient sequence of items from a pick list to reduce total travel distance in low-level manual picking systems. Weidinger [57] developed a nearest-neighbour-based metaheuristic to enhance picker routing in warehouses with mixed-shelf storage. His method assesses the impact of various mixed-shelf scenarios, providing comparative insights with traditional storage policies to optimize urgent order assembly.

While the studies presented in Table 1 demonstrate a variety of metaheuristic approaches applied to order-picking routing problems, our study stands out by explicitly addressing the challenge of random storage and the operational complexity associated with multiple products per storage position. In contrast to the predominant literature focused on more predictable and dedicated storage scenarios, our work introduces an innovation by adapting the FTSP to address the specificities of random storage. This innovation is developed with GRO, a genetic algorithm to optimise picking routes in complex storage environments. Thus, the main contribution of our study lies in the practical application of GRO in a real industrial scenario, using authentic warehouse data, an approach rarely explored in previous research. This validates our method's effectiveness in an operational context and offers insights for managing warehouses facing similar challenges.

An important aspect to consider in the context of travelling and picker routing is the type of picker involved, whether human or robotic (e.g., AMR - automated mobile robot). Our study specifically focuses on human order pickers operating within a complex warehouse environment. This distinction is crucial, as the optimisation strategies and challenges can differ significantly between human and robotic pickers. Human pickers provide the flexibility and adaptability to navigate the random storage and varied product types typical of the warehouse we studied. This focus aligns with studies such as Battini et al. [66], which highlight the ergonomic and efficiency challenges faced by human pickers, and Grosse et al. [67], which emphasize the importance of optimising travel distance to reduce physical strain and improve productivity. Additionally, most of the studies analysed in Table 1 indicate that the work is performed manually and the distance travelled factor is the most analysed measure in these cases, reinforcing the relevance of focusing on human order pickers. Our research aims to develop practical solutions that enhance route efficiency and operational performance in real-world warehousing scenarios by addressing human pickers' unique needs and capabilities.

IV. SOLUTION APPROACH

In this section, we summarise the solution approach proposed by the study, focusing on GRO, a metaheuristic methodology designed to efficiently address the complexities of warehouses with random storage and multiple products per storage position. GRO is based on GA, a widely used method to solve optimisation problems [68].GA was selected for the development of the GRO due to its adaptability to deal with the complex and randomised storage environments present in our case study.

Thus, GRO starts with a population of candidate solutions, evolving through selection, crossover and mutation processes to meet the challenge of optimising picking routes for warehouses with random storage and multiple products per location. This iterative method allows it to adapt dynamically to the complexities of storage and classification demands, continually refining the solutions [69].

Figure 2 illustrates the process of our solution approach, starting with data gathering on the warehouse layout, start and end points of the route (i.e., depot) and list of items to be collected. This is followed by a phase of technical adjustments and pre-processing of this data to prepare it for the GRO application. After running GRO, the process is completed by generating a detailed picking report and a visual map highlighting the storage positions to be visited to optimise the picking route in the warehouse. In the following sections, we detail each stage of this approach, clarifying how GRO dynamically adapts to the storage complexities and picking requests to reduce picking distances.

Table 2 presents a detailed nomenclature essential for understanding GRO operating parameters. Each algorithm component is discussed in the following sections, highlighting aspects such as initial population generation, genetic operator selection and convergence strategies.

A. INITIAL SETUP

Identifying a finite set of data points is essential in adapting our approach to various warehouse layouts. These data

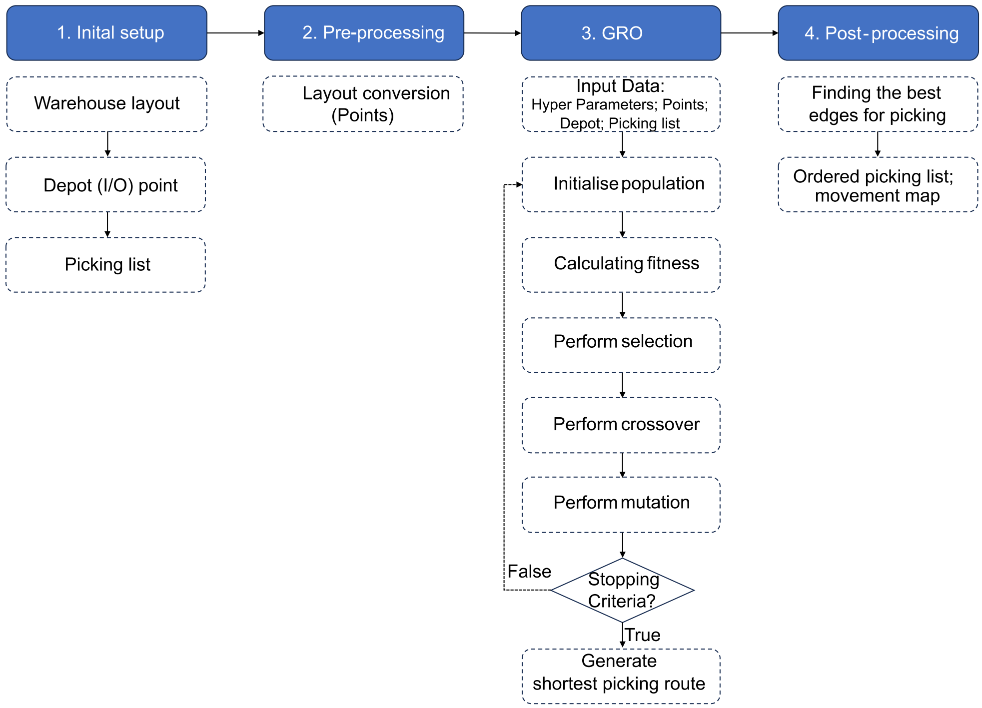


FIGURE 2. Flowchart of the proposed approach.

points must uniquely define the warehouse layout and be mathematically expressible, allowing algorithmic processing. To represent the problem, which aims to minimise the total distance travelled by pickers, it is necessary to know precisely the distances between all pairs of storage locations (or storage bins) within the warehouse under study [70]. The warehouse has several parallel aisles intersecting with cross aisles. The horizontal aisles are designed to be narrow, enabling operators to access items from both sides. The shortest picking route problem must be solved to calculate the distances between various locations. It is important to note that pickers can navigate the side aisles and the central aisle when transitioning between different locations, often resulting in varying distances.

To implement GRO, we defined a set of preliminary steps. The first stage involves a detailed configuration of the warehouse layout, precisely defining shelf dimensions, aisles, and the exact location of each picking bin. Subsequently, the start and end points of each picking cycle are determined, typically based on the warehouse entry/exit locations, to ensure an efficient picking route and minimise turnaround time. Finally, the list of products to be picked needs to be compiled and optimised, considering the ideal sequence to minimise total travel distance, taking into account the location of each product within the warehouse. From there, using the pick list established from the company's Warehouse Management System (WMS), we associated each product picking location with a specific point. This allowed us to assign distances to each position to be visited on the pick list. With this approach, we calculated the total distances required to collect all the products listed, resulting in an efficient and detailed analysis of the necessary route for the picking tasks.

In developing GRO, we adapted metaheuristic strategies, including the FTSP concept, to address warehousing challenges with random arrangement and multiple products per position. The integration of FTSP allows GRO to effectively manage the complexity of visiting "families" of products, ensuring a solution adapted to the specific needs of our warehouse and highlighting our study's practical contribution to warehouse management.

B. PRE-PROCESSING

In the pre-processing stage, the regular structure of the warehouse was modelled on a Cartesian plane, making it easier to map shelves, aisles and storage areas accurately. This representation makes it possible to assign specific coordinates to each element within the warehouse, representing the location of each product by its coordinates (x_p, y_p, z_p). Here, x_p and y_p refer to the product's position on the x and y axes, respectively, while z_p refers to the storage system level where the product is located. This approach allows an accurate representation not only of the physical space of the warehouse, but also facilitates the identification and calculation of the most efficient paths between products, taking into account the three-dimensional layout of the environment.

In addition to spatial modelling, we established an initial representation of the stock that mirrored the configuration found in the company during data collection. This detailed representation included the identification of storage shelves

TABLE 2. Nomenclature.

Category	Parameters	Description
Sets	$p \in P$	Set of products
	$t \in T$	Set of position
	$i \in I$	Set of individuals
	$n \in N$	Set of individuals for the next generation
	$g \in G$	Set of positions contains the product p
	$v \in V$	Set of items p to be picked at points t
	$b \in B$	Set of intersections between aisles
	$r \in R$	Set of locations to be visited
Structures	$d \in D$	List of positions defined for p
	$h \in T$	List of ordered points in i
	`fitness`	Vector of fitness value for each individual
GA Parameters	α	Population size
	β	Generations size
	γ	Crossover probability
	ω	Mutation probability
	λ	Tournament size
Variables	o	An individual (ordered list)
	z	Generation counter
	a, b	Pair of individuals for crossover
	s_index	List of indices from 1 to $\|n\|$
Function	$w(T, V, P, \alpha)$	Initialises the population I
	$f(I)$	Fitness of the individual $i \in I$
	$s(I, \lambda)$	Selects of the individual $i \in I$
	$c(N, \gamma, \alpha)$	Applies crossover in $n \in N$
	$m(N, \omega)$	Applies mutation in $n \in N$
	$GA(T, P, \alpha, \beta, \gamma, \omega, \lambda)$	Performs the genetic algorithm
	$Distance(point1, point2)$	Calculates the distance between points

holding multiple products, a distinctive feature of our warehousing challenge. By simulating the current arrangement and quantity of products in each location, we created an overview of the stock, which is essential for the subsequent optimisation of the picking process. This initial study captures the diversity and distribution of products within the warehouse. It serves as the basis for the GRO algorithm to identify optimised pick routes that minimise the travel required to reach all the items on the pick list.

C. GENETIC ROUTE OPTIMISATION

The pseudo-code outlines a portion of GRO, as shown in Algorithm 1, an algorithm for generating a picking route. In the initial phase, a population of candidate solutions is represented as individuals, where each individual corresponds to a possible picking route. Each individual's fitness is calculated using a fitness function, which considers the distance travelled around the warehouse during the picking process. Subsequently, the algorithm develops the population in three main steps: selection, crossover and mutation.

In the selection step, a set of individuals is chosen based on their fitness, using a selection mechanism favouring individuals with better fitness values. Next, in the crossover step, pairs of individuals are combined to produce new individuals, which can be the offspring of both parental solutions. The algorithm controls the crossover rate through the parameters γ and α. Finally, in the mutation step, some new individuals are randomly modified to introduce diversity into the population. These steps are repeated over several generations (β) to find the best picking route. The final result is the individual with the lowest fitness in the population after completing the iterations. The GRO algorithm is a promising approach to efficiently solve the challenging problem of warehouse product picking routing,

Algorithm 1 Genetic Route Optimisation

Function $GA(T, P, \alpha, \beta, \gamma, \omega, \lambda)$

$I \leftarrow w(T, V, P, \alpha)$ `// Initialise population`

for $z \in \{1, \ldots, \beta\}$ **do**
- `fitness` $\leftarrow f(I)$ `// Calculating fitness`
- $N \leftarrow \emptyset$ `// Initialise new population`
- $N \leftarrow s(I, \lambda)$ `// Perform selection`
- $N \leftarrow c(N, \gamma, \alpha)$ `// Perform crossover`
- $I \leftarrow m(N, \omega)$ `// Perform mutation`

end

Result: $\{f(I)\}$ `// best individual in` I

combining natural selection, recombination and mutation to find increasingly better solutions over generations.

The following topics analyze the functions used in the algorithm. We will explore selection, crossover and mutation operations, fitness function, and best individual selection, highlighting how these steps combine to find optimised solutions in complex warehouse picking routing problems.

1) INITIAL POPULATION GENERATION

The Algorithm 2 is called "Initialise Population" and is critical in the GRO algorithm in addressing warehouse product picking routing problems. This function generates an initial population of candidate solutions, each representing a potential picking route.

Algorithm 2 Initialise Population

Function $w\ (T, V, P, \alpha)$
- $I \leftarrow \emptyset$ `// Initialise a population`
- $V^* \leftarrow$ `Permutation` (V) `// Change the order of V`
- **while** $|I| < \alpha$ **do**
  - $i \leftarrow \emptyset$ `// Initialise an empty individual`
  - **for** $p \in V^*$ **do**
    - $G_p \leftarrow \emptyset$ `// A set of points is initialised`
    - **for** $t \in T$ **do**
      - **if** *the product p is available at t* **then**
        - $G_p \leftarrow G_p \cup \{t\}$ `// Add t to` G_p
      - **end if**
    - **end for**
    - $i \leftarrow i \cup$ `Sample`$(G_p, 1)$ `// A sample in` G_p
  - **end for**
  - $I \leftarrow I \cup \{i\}$ `// Add i to I`
- **end while**

Result: I

The algorithm begins by initializing an empty set, denoted as I, which will hold the population of individuals. To introduce diversity in the initial solutions, the order of product set V is randomised by applying a permutation operation, identified as `Permutation` (V). This randomisation step is often used in algorithms to introduce randomness and diversity, which can be beneficial in exploring different solutions, especially in optimisation problems [71].

Next, an empty individual, represented as i, is initialised within this loop. For each product p in the randomized order of V^*, a set G_p is initialised to store possible points (locations) where product p is available for collection. The algorithm iterates over the points slots t in set T (representing available picking positions) and checks if product p is available at position t. If it is available, the position t is added to the set G_p. A sample operation, denoted as `Sample`$(G_p, 1)$, randomly selects one point from the set G_p and adds it to the individual i. This step represents the selection of a specific location to collect each product in the route.

Figure 3 illustrates the strategy adopted to generate the initial population in GRO. Each family represents a set of available storage positions within the warehouse where the specific product p can be found. This approach allows to identify all possible locations for each item. Once these families are formed, the next step is to generate individuals for the algorithm's initial population, where each individual's gene symbolises a specific collecting position for a product. The selection of the position for each product on the picking route is carried out randomly, guaranteeing diversity in the initial solutions and promoting a broad exploration of the search space.

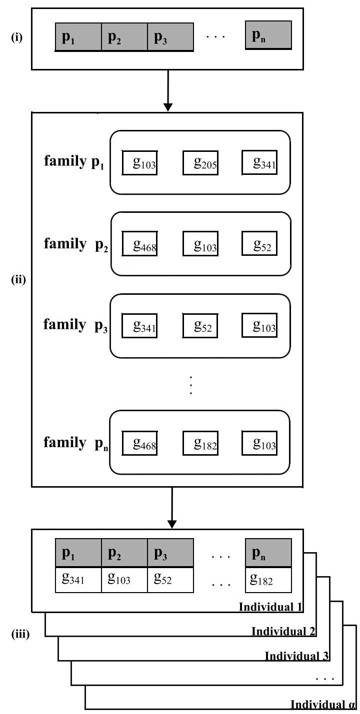


FIGURE 3. Strategy for generating the initial population in the GRO.

2) FITNESS VALUE EVALUATION

The "Fitness" function, shown in the Algorithm 3, primarily calculates the fitness of each route (represented by i) within the population I. The fitness value measures the performance of a given picking route concerning the defined objectives and constraints of the problem and influences the selection of routes for the next generation.

The "Fitness" function, as outlined in Algorithm 3, calculates the fitness of each route in a set of candidate routes I. The fitness value measures a route's efficiency in minimising the travel distance for product collection. This calculation considers the positions of consecutive points in the route, determining whether they are in the same aisle or different aisles. The Distance function determines the

Algorithm 3 Fitness Function

```
Function f (I)
foreach i ∈ I do
    fitness_i ← 0            // Initial fitness value
    foreach h ∈ {1, 2, ..., |i| − 1} do
        point1 ← i_h
        point2 ← i_{h+1}
        dist ← Distance(point1, point2)
        fitness_i ← dist // Add distance to fitness
    end foreach
end foreach
Return fitness
```

distance between two points, computed as follows. If two points are in the same aisle, the distance is calculated using Eq. 8:

$$\text{Distance} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{8}$$

where (x_1, y_1) and (x_2, y_2) are the coordinates of the points in the same aisle.

If the points are in different aisles, the distance is calculated as the sum of the vertical distance (dy) and the minimum horizontal distance (dx) to reach either the aisle's right or left end, using Eq. 9 and Eq. 10, and the result is shown in Eq. 11:

$$dy = |y_2 - y_1| \tag{9}$$

$$dx = \min(|x_\mathrm{r} - x_1| + |x_\mathrm{r} - x_2|, |x_1 - x_\mathrm{l}| + |x_2 - x_\mathrm{l}|) \tag{10}$$

$$\text{Distance} = dy + dx \tag{11}$$

where x_r and x_l represent the coordinates of the right and left ends of the aisle, respectively.

The function `CalculateDistance` determines the fitness of a route, directly influencing the efficiency with which the GRO algorithm identifies and selects optimal routes for subsequent genetic operations.

Suppose two consecutive points belong to different aisles. In that case, the fitness value accounts for the vertical distance between them and the minimum horizontal distance to reach either the right aisle (x_r) or the left aisle (x_l). The fitness values computed for all candidate routes play a key role in the following stages of the GRO algorithm. Routes with lower fitness values, indicative of superior performance in minimising travel distance, are prioritised during subsequent genetic operations such as crossover and mutation.

3) SELECTION

Algorithm 4, called "Selection", plays a fundamental role in the evolutionary process, as it is responsible for selecting a subset of individuals from the current population to form the basis of the next generation of potential solutions.

Algorithm 4 Selection

```
Function s (I, λ)
N ← Ø     // Initialises a population N
D ← Ø          // Initialises a list of positions D
for i ∈ I do
    D ← Sample (I, λ) // The tournament is I
    o ← argmin_{d∈D}{f(d)}      // Select d with shortest picking route
    N ← N ∪ {o}             // Add o to N
end for
Result: N
```

This algorithm takes two input parameters: the current population I and parameter λ, which determines the size of the tournament selection group. The function begins by initialising two empty sets, N and D, which are used to store the individuals selected for the next generation and the positions of individuals in the tournament selection group, respectively. The function creates a tournament selection group for each i in the current population I by randomly sampling λ individuals from I. This sampling process is accomplished using the `Sample`(I, λ) operation.

The algorithm identifies the individual o within each tournament selection group with the optimal performance, which in this context is the shortest picking route, determined by $argminf(D)$. Here, $f(D)$ represents the distance values of the individuals within the group D. The individual o, having the optimal shortest picking route, is then included in the next generation (N), contributing to the evolution of the population in the subsequent iteration.

4) CROSSOVER AND MUTATION OPERATION

Algorithm 5 is responsible for recombination or crossover, where pairs of individuals are selected from the current population to produce offspring that inherit characteristics from both parents.

This algorithm uses three input parameters: the current population N, a crossover probability parameter γ, and a population size parameter α, which determines the size of the next generation. Initially, the function uses a loop that iterates from $i = 1$ to $\frac{\alpha}{2}$. In each iteration, it selects a pair of individuals (a, b) randomly from the current population I using the `Sample`$(I, 2)$ operation.

For each selected pair of individuals, a probability check is performed using the `rand`$(0, 1)$ function, comparing the result to the crossover probability γ. A crossover occurs if the random value is less than γ. This probability check introduces a level of randomness into the crossover process. The function proceeds with the crossover operation when the crossover probability condition is met. It first determines a crossover point h^* as half of the length of individual a. Then it creates two new individuals, Child01 and Child02,by combining the

Algorithm 5 Crossover

```
Function c (N, γ, α)
for i = 1 to α/2 do
    (a, b) ← Sample (I, 2)        // Select a pair from I
    if rand (0, 1) < γ then
        h* ← [|a|/2]
        a ← [a_1, ..., a_{h*}, b_{h*+1}, ..., b_{|V|}]    // Child01
        b ← [b_1, ..., b_{h*}, a_{h*+1}, ..., a_{|V|}]    // Child02
    end if
    N ← N ∪ {a, b}    // Update a and b in N
end for
Result: N
```

Algorithm 6 Mutation

```
Function m (N, ω)
foreach n in N do
    h* ← [|n|/2]
    if rand (0, 1) < ω then
        s_index, x ← randint(h*, |n|)    // Exch.index
        aux ← n[x]
        n[x] ← n[s_index]    // Swap gene values
        n[s_index] ← aux     // Update n
    end if
end foreach
Result: N
```

first part of one parent with the second part of the other parent. This swapping of genetic material between parents generates two offspring that inherit characteristics from both parents. The offspring individuals, a and b, are added to the next generation, N, effectively replacing the worst parents in the population.

Choosing a crossover point among the individuals is based on exploring new areas of the solution space and exploiting existing solutions [72]. This half-point crossover strategy aims to effectively combine the attributes of the parents to produce offspring that inherit significant characteristics from both, increasing the chances of generating high-quality solutions, allowing an equitable distribution of route parts to the offspring, favouring the retention of beneficial point sequences that may have been established in previous generations [73]. In addition, this approach helps maintain genetic diversity within the population, avoids premature convergence toward local optima, and guarantees a broad exploration of the search space [74].

Following the crossover process, the genetic algorithm proceeds to the mutation phase, as described in Algorithm 6. Mutation is another mechanism for introducing diversity and exploring new genetic material within the population. This phase ensures that the genetic algorithm does not become stuck in local optima and continues to search for novel and potentially improved solutions.

The Mutation function takes two essential input parameters: the population N and the mutation probability parameter ω. The function prepares for potential mutation for each n in the population N. It calculates a midpoint h^*, representing half the length of an individual's genetic representation. To introduce diversity, a random probability check is performed for each individual. Mutation is initiated if a randomly generated value from `rand`(0, 1) is less than the mutation probability ω. This probabilistic approach ensures that not every individual undergoes mutation, preserving a balance between exploration and exploitation. Mutation involves selecting two indices, s_{index} and x, within the individual's genetic representation. These indices determine the positions where gene values are exchanged. The swap operation effectively alters the individual's genetic makeup. Subsequently, the mutated individual is updated within the population N.

The implemented mutation strategy is designed to inject diversity into the population and avoid premature convergence toward sub-optimal solutions [73]. This approach ensures that the metaheuristic continues to explore new possibilities for choosing solutions, even at advanced stages of evolution, when genetic variability tends to decrease, providing an effective mechanism for escaping local minima and allowing the exploration of unvisited areas of the search space [72]. According to Engelbrecht [74], this strategy makes it possible to balance stability and innovation for new solutions, ensuring that changes are preserved while new route configurations are tested.

Upon completing the GRO iterations, the final step is determining the best solution among the individuals. This solution is identified by finding the individual within the population with the shortest picking route, calculated using the $f\ (I)$ function. In this context, shorter distances correspond to superior solutions to the optimisation problem. The individual with the shortest picking route, obtained through the $argmin\{f(I)\}$ operation, represents the optimal picking point sequence. This sequence provides a clear and practical route for warehouse product picking, efficiently minimising travel distances and improving overall operational efficiency.

D. POST-PROCESSING

After applying GRO to optimise the product-picking route in the warehouse environment, the next step was transforming the GRO results into a practical route for the operator. We generated the graph edges to represent the connections between the picking locations previously identified as key points. The points in GRO served as nodes in the graph while we calculated the edges to represent the distances and the optimal order in which the points should be visited. This approach allowed us to create a targeted route for the

TABLE 3. Warehouse parameters.

Parameter	Value	Units
Number of levels	2	Unit
Number of racks	26	Unit
Total vertical Aisles	3	Unit
Total parallel Aisles	17	Unit
Capacity of a storage unit	18	Unit
Total storage units	2040	Unit
Products range per pick list	15 to 27	Products

TABLE 4. Values of GRO parameters.

Parameter	Value	Units
Population size (α)	1000	Individual
Genetic iterations (β)	1000	Iterations
Tournament selection size (λ)	3	Individual
Crossover rate (γ)	80	%
Mutation rate (ω)	10	%

collection tasks, optimising the process of picking products from the warehouse.

V. ANALYSING THE PRACTICAL CASE

After detailing GRO, we expanded our analysis to a broader set of tests. This section presents a comprehensive analysis covering large-scale tests to evaluate the proposed algorithm's effectiveness and adaptability in different scenarios and data sets. The results of these tests offer a more comprehensive view of the method's performance and viability in different contexts.

A. WAREHOUSE CONFIGURATIONS AND METAHEURISTIC PARAMETERS

The warehouse operational characteristics play a key role in determining the effectiveness of routing and picking strategies. Bidirectional aisles characterise the warehouse configuration and a regular rectangular architecture with three blocks and transverse aisles, as shown in Figure 3. This configuration was converted into parameters, shown in Table 3, in which the information on aisles and shelf levels was converted into points on the Cartesian plane.

Thus, the GRO algorithm aims to optimise the first two storage levels, to reduce the distance travelled by the selection operators, where each point on the selection list symbolises a specific destination. Also considered is the configuration of the warehouse's longitudinal aisles, designed to allow easy access to items from both sides. This analysis focuses on operations with one operator at a time, minimising concerns about congestion. Another relevant aspect is operator mobility, who can change their route in the aisles. The procedure starts and ends at the warehouse base, with the operator returning after completing the picking of items. The aisles are bidirectional, and the warehouse architecture is characteristically regular, usually with a rectangular layout and parallel longitudinal aisles.

Each picker receives a picking list containing 15 to 27 products, in which each position retrieved from the WMS must be visited. Armed with specially designed trolleys that support the collection of items, operators head to the warehouse to start picking. These trolleys are specifically developed to carry cartons, enhancing the efficiency and organization of the picking process. It is worth noting that there is no specific heuristic that dictates the path to be taken by the picker when selecting products. The route is based on the operator's intuition and experience, allowing flexibility in the picking process and enabling dynamic adaptations depending on the layout of the products in the warehouse. However, this flexible approach can result in two main challenges. First, the lack of a predefined strategy can generate significant variations in picking times, leading to inconsistencies in operational efficiency. In addition, reliance on operator intuition can lead to sub-optimal routes, increasing the likelihood of rework or travelling longer distances, potentially having a negative impact on operational productivity.

In addition, the list of products to be picked is a parameter that guides the picking process and the efficiency of the GRO algorithm. For each order, the list specifies a series of picking locations in the warehouse and the corresponding products that must be picked. Each entry in the list details the sequential order of picking, starting with the first item and progressing to subsequent products. This order defines the formulation of the routing problem and evaluates the algorithm's effectiveness in minimising the total route and picking time. The complete list contains 15 to 27 items, each associated with a specific location and order number, including at least four different positions to be visited, with several products sometimes located in the same position, up to a maximum of 27 different positions. Analysing these picking lists allows us to test and validate the applicability of our GRO algorithm in practical scenarios, taking into account variations in product locations and picking patterns.

The effectiveness of GRO in our study depends on the precise calibration of parameters detailed in Table 4, we present the final parameters that emerged from this calibration process. We adjusted the parameter values through an iterative calibration process to optimise the algorithm's performance, seeking an ideal balance between exploring new solutions and exploiting promising routes. These include population size (α), which determines the number of candidate routes considered in each generation; the number of genetic iterations (β), which reflects the number of selection, crossover and mutation cycles carried out; the size of the tournament selection (λ), which directly affects the selective pressure when choosing parents; as well as crossover (γ) and mutation (ω) rates,which are fundamental for defining the frequency of genetic operators. This calibration process ensured that GRO could deal efficiently with the complexity and variables of storage environments in the context of our study.

B. GRO PERFORMANCE

To evaluate the effectiveness and adaptability of GRO in the real operational environment under study, we conducted an empirical analysis using real picking data. This evaluation focused on a diverse set of 51 picking lists, each reflecting a unique picking scenario in the warehouse. The lists were extracted from the company's warehouse management software (WMS), which is under investigation and covers various picking situations. The selection was made using a stratified random sampling method to accurately represent the diversity of picking scenarios in the warehouse. This method involved categorizing the picking lists based on key variables, such as the size of the order, the diversity of products and the complexity of the storage location, and then randomly selecting samples from each category. This approach ensured that the 51 chosen lists encompassed the range of operating conditions faced by the company, making the sample representative of overall picking operations. By covering this range of variables, the selected samples provided a comprehensive view of the company's daily picking operations, allowing for detailed comparative analysis and ensuring that the conclusions and optimisations derived from the GRO could be generalized to improve the efficiency of the overall picking process.

To ensure our statistical comparisons' validity and subsequent analyses' suitability, we performed normality tests on the company's current picking distances and those obtained by the GRO, using the Shapiro-Wilk test. These tests aimed to determine whether the data followed a normal distribution, a crucial assumption for many parametric statistical tests. Establishing the normality of the data allows us to apply these tests confidently and accurately compare the GRO's performance with current picking methods. We obtained a statistical value of 0.9727 and a p-value of 0.2850 for the current distances, indicating a normal distribution. The statistical value of the distances generated by GRO was 0.9589, with a p-value of 0.0752, suggesting normality. Histograms and comparative density lines (Figure 4) represent these results. The analysis showed that current distances range up to around 100 metres. At the same time, those optimised by GRO are concentrated in a narrower range, indicating a trend toward shorter distances (Figure 4a). The Kernel Density Estimate (KDE) applied to the distances (Figure 4b) showed a steeper GRO curve, with a higher density for distances up to 50 metres, reflecting GRO's effectiveness in reducing collection distances.

The results of the descriptive statistics, as shown in Table 5, reveal important insights into the distances. The company's average distance practised was approximately 79.63 m, with a standard deviation of 41.37 m, indicating considerable dispersion around the average. The shortest picking route recorded was 11.05 m, while the longest was 205.84 m. Similarly, the distances obtained by the genetic algorithm had an average of 62.52 m, with a standard deviation of 32.35 m, and ranged from 12.03 m to 171.04 m. Both data sets showed coefficients of variation of around 51.95% and 51.75%, respectively, suggesting a similar relative variation regarding their averages. After implementing GRO, the average collection distance was reduced from 79.63 metres to 62.52 metres, increasing efficiency by approximately 21.48%. This improvement highlights the effectiveness of GRO in reducing collection distances, emphasising its importance for faster and more economical operations. A paired t-test was conducted to check for statistically significant differences between the actual distances and those optimised by GRO. The results indicated a significant difference (t-statistic = 9.1355, p-value = 0.000), confirming the statistical discrepancies between the mean distances.

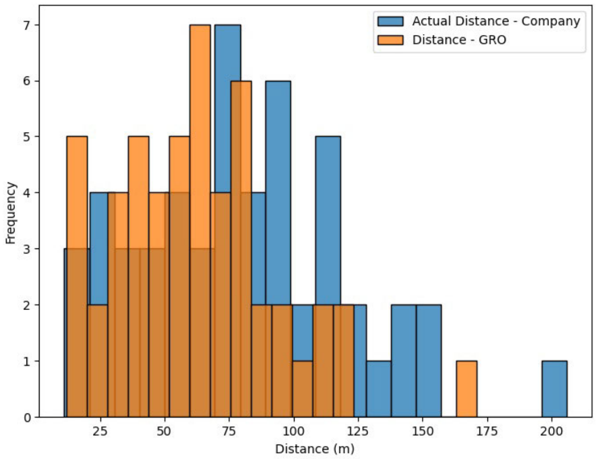
TABLE 5. Descriptive statistics.

Statistical Measure	Distance - Actual (m)	Distance - GRO (m)
Mean	79.63	62.52
Standard Deviation	41.37	32.35
Min	11.05	12.03
25%	49.75	40.70
50%	73.71	61.27
75%	108.23	81.01
Max	205.84	171.04
Coefficient of Variation	51.95%	51.75%

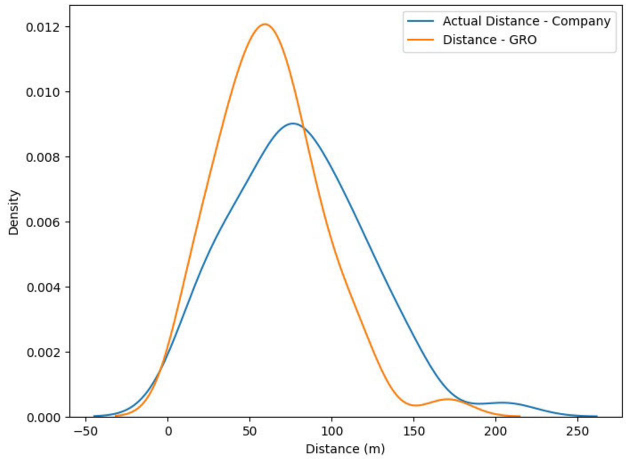
When analysing the descriptive statistics, we noticed a dispersion in the actual distances practised by the company and those optimised by GRO, indicated by the standard deviation values. Figure 5 compares the actual distances with those optimised by GRO, indicating GRO's ability to reduce the total picking distance in various scenarios. Figure 5(a) shows the dispersion of the points, indicating improvements in most of the collection lists. However, in others, the gains are more modest, reflecting that factors can influence the algorithm's effectiveness. This point-by-point analysis makes it possible to assess the applicability of GRO. Figure 5(b) compares, through overlapping bars, the distances used by the company and the results generated by GRO for each pick list, showing that the reduction observed was 56.77 metres, while the smallest significant reduction, excluding negative values, was just 0.56 metres. On average, GRO's reduction in collection distances was approximately 17.11 metres.

Figure 6 highlights the differences in picking distances before and after applying GRO. The distances optimised by GRO show a narrower concentration around the median (Fig. 6(a)), suggesting improved route consistency and efficiency. While the accurate distances vary widely, reaching a significant maximum of 205.84 metres, the GRO-optimised distances have a narrower range, with a maximum of 171.04 metres. Comparing quartiles between both data sets highlights the improvement, with GRO producing shorter distances at each reference percentage point, optimising picking routes. Figure 6(b) represents the ascending linear trend, highlighting the effectiveness of GRO in reducing picking distances compared to the company's previous practices for varying pick list sizes.

The relationship between the size of the picking lists and the execution time to generate GRO results was analysed

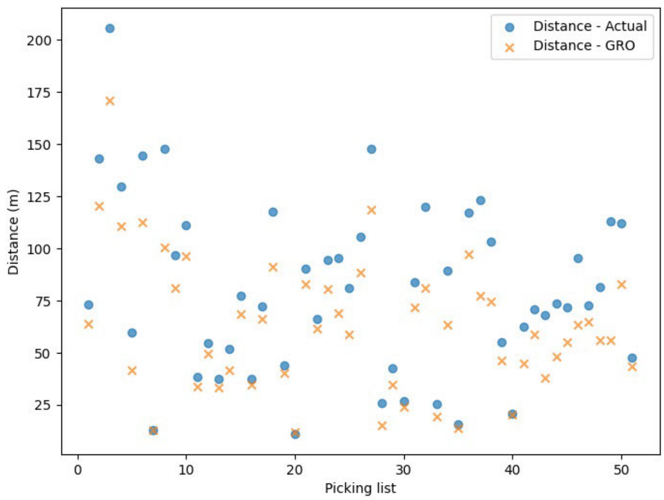


(a) Histogram of the distances practised by the company and GRO.

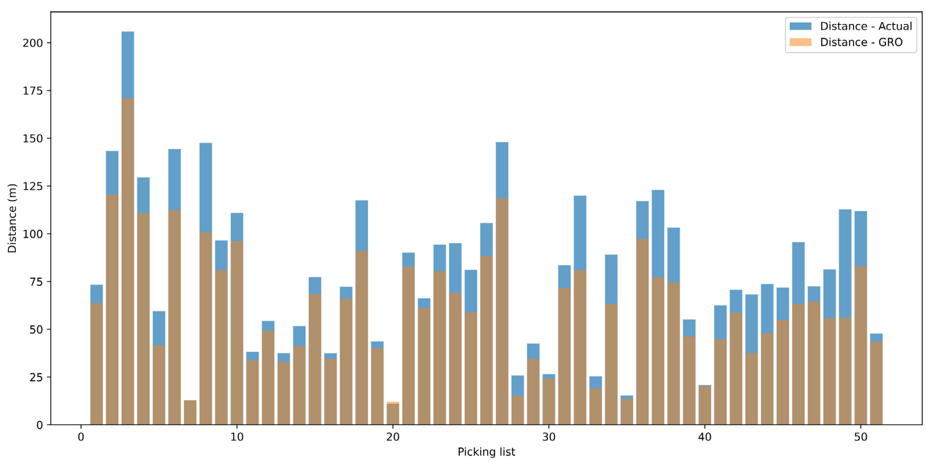


(b) Density line.

FIGURE 4. Distribution of distances with histogram and comparative density.



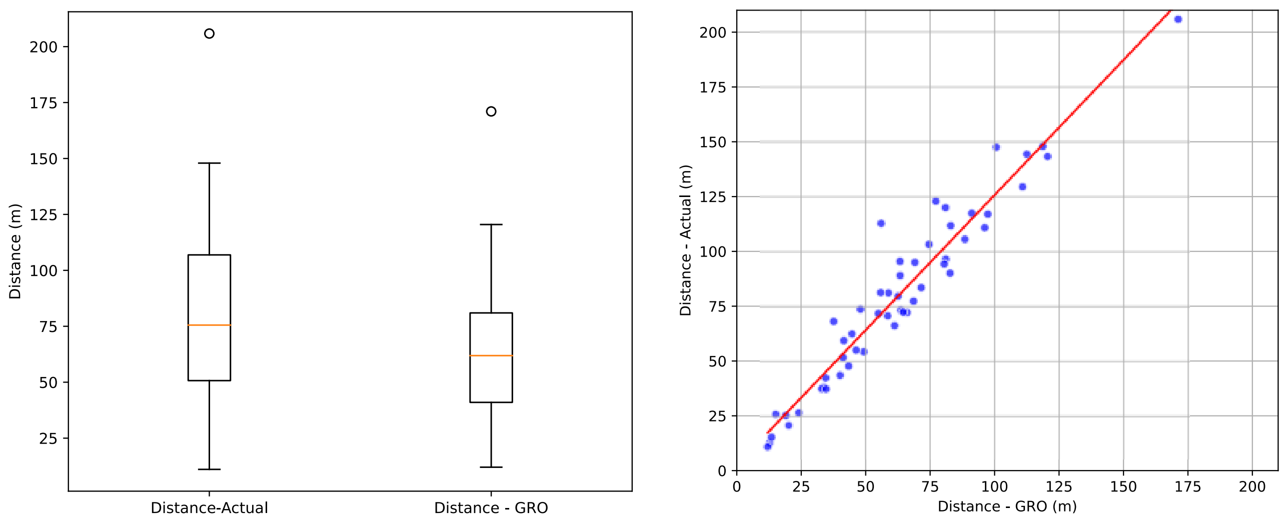
(a) Bottom chart of the distances practised by the company and GRO.



(b) Bar chart of the distances practised by the company and GRO.

FIGURE 5. Comparison of the distances practised by the company and GRO.

using a notebook with a 1TB hard drive, a 10th-generation Core i9 processor and 16 Gb of memory. The results showed a variation in response time with list size: for 23 items, the minimum time recorded was 112.14 seconds, while for



(a) Boxplot of the distances practised by the company and GRO.

(b) Scatter the distances practised by the company and GRO.

FIGURE 6. Boxplot and Scatter of the distances practised by the company and GRO.

27 items, the maximum was 157.20 seconds. The average time for the lists tested was around 135.69 seconds, indicating that longer pick lists increase processing time.

The comparative analysis between the picking routes generated by GRO and current practises demonstrates the potential of GRO to optimise logistics processes, showing an improvement in operational efficiency through a reduction in average collection distances. However, the similarity in the coefficient of variation between both data sets underlines a constant functional variability influenced by factors such as product diversity and stock dynamics, which persists regardless of the methodology applied. Therefore, despite the advances made by GRO, its effectiveness must be considered in conjunction with these variabilities, emphasising the need for adaptive and flexible strategies for effective warehouse management.

VI. DISCUSSIONS

A detailed analysis of the GRO performance revealed an optimisation of the picking routes within the warehouse studied, evidenced by a substantial reduction in average distances travelled to collect items. Compared to current practises, GRO reduced picking distances by an average of 21.48%. This result validates GRO's effectiveness in optimising picking routes and suggests an operational improvement that could reduce picking times and costs associated with movement within the warehouse. Reducing picking distances has direct implications on warehouse operational efficiency. First, there are immediate savings in the time operators spend picking items, allowing resources to be reallocated to other critical logistics activities. Second, reducing the physical effort required of operators can contribute to less fatigue and, potentially, a decrease in error rates, increasing the accuracy of picking operations.

GRO's innovation in integrating an extended version of FTSP concepts into its development process is advantageous in tackling the challenges of warehouses with random stock and multiple products per position. By grouping picking locations into families based on criteria such as product category, FTSP allows for a more structured approach to optimising picking routes. This methodology makes it easier to locate and pick products within a complex storage environment and ensures that pick routes are logically organised to minimise unnecessary movement. The ability to dynamically adapt picking routes in response to changes in product location or order composition demonstrates significant operational flexibility, essential for warehouses facing regular fluctuations in stock and demand.

The process of implementing GRO-optimised results begins with the generation of picking lists by the WMS, reflecting the warehouse's daily operational demands. The manager then forwards these lists to GRO, which applies its metaheuristics to analyse and optimise the picking routes. This procedure results in up-to-date picking lists and detailed maps, which outline the picking points sequentially. This approach significantly facilitates the task of the pickers, providing clear guidance throughout the warehouse and ensuring a more efficient and systematised picking operation.

This transformation in operational efficiency is particularly critical in contexts where pre-defined handling patterns are non-existent, as observed in the company under study. The lack of a structured routing system often resulted in inefficiencies and picking errors, reducing operational productivity and accuracy. Figure 7 illustrates a section of the optimised route map generated by GRO, highlighting the picking positions used. The company adopts a pattern to identify a stocking position. For example, a typical representation would be J-10-21. Where: J indicates aisle J, 10 indicates that it is in the tenth compartment of aisle J,

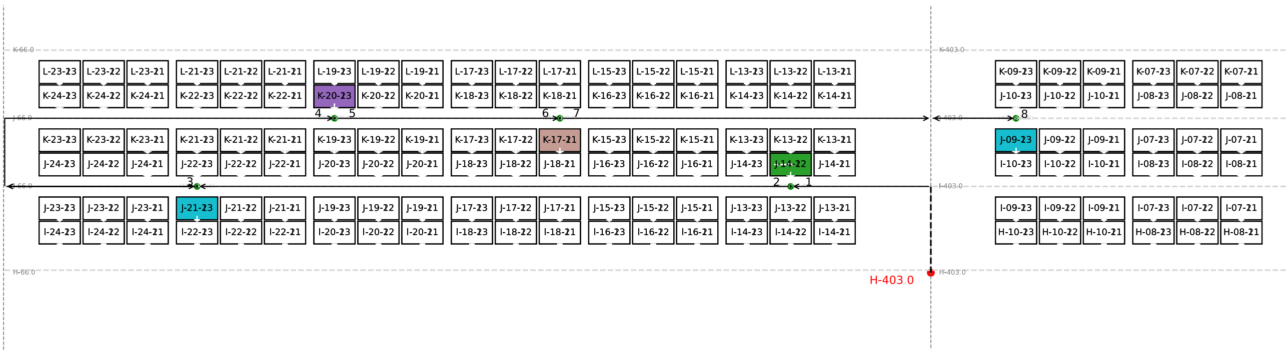


FIGURE 7. Section of the product picking map.

2 indicates that it is on level 2 (second floor), and 1 indicates that this is the first location in that compartment and on that floor (there can be several locations in the same compartment and on the same floor).

Figure 7 illustrates the optimised separation route generated by GRO. Starting from the entry point, highlighted in red, the GRO algorithm applies genetic principles to minimise the distance travelled in an ordered sequence of locations numbered 1 to 8. GRO identifies a near-optimal path that systematically reduces unnecessary movement and maximises efficiency by iteratively refining the possible routes. The sequence incorporates support points strategically positioned along the side and centre aisles, designed to make navigation easier for operators. Each numbered point corresponds to the precise order of product collection, structuring the process into a logical flow that is predictable and efficient. As the company currently operates without a standardised picking sequence, this optimisation demonstrates GRO's ability to transform warehouse operations by applying data-driven picking processes to improve real-world efficiency.

When comparing the GRO computational time with similar studies, we noticed that, despite being applied to instances involving up to 2000 storage positions, our algorithm maintained times in the 135 seconds range, in line with the results of [56], [63], [64], and [65] for similar capacities. Interestingly, studies that looked at warehouses with more than 5000 positions, such as those by [58], [60], and [61], reported shorter computational times. However, these studies implemented complementary computational efficiency strategies, such as ACO, which, when combined with GA, can significantly reduce search times by dynamically adjusting the search space size for more promising solutions. This approach suggests a potential avenue to further optimise GRO performance in future research, adapting to increasing complexity without compromising operational efficiency, especially in large-scale warehouses.

In addition to the time efficiency highlighted, GRO demonstrates its quality through its robustness and flexibility when dealing with significant variations in the size and composition of collection lists. While previous studies, such as those by [45], [53], [55], and [62], predominantly focused on scenarios with static or limited collection parameters, GRO was designed to dynamically adapt to a wide range of operational scenarios, reflecting the real complexities of warehousing environments. This ability to adjust in real-time increases GRO's efficiency and highlights its applicability in warehousing environments that face constantly changing picking demands, an everyday reality in modern logistics.

The application of the GRO in the warehouse environment highlights the critical interplay between engineering principles and management practices. We addressed a technical challenge by optimising picking routes while providing significant managerial insights. Reducing travel distance translates into increased operational efficiency, lower labour costs, and improved worker productivity, showcasing the tangible benefits of integrating advanced engineering solutions into warehouse management strategies. Additionally, the visual solutions generated by the GRO enabled better oversight and planning, allowing managers to identify bottlenecks and areas for further optimisation. This case study underscores the necessity of combining engineering innovations with strategic management to enhance overall warehouse performance, demonstrating the practical significance of research at this intersection.

Tests of GRO have shown it to be an effective strategy for optimising picking routes in a specific warehouse scenario, resulting in improved operational efficiency. However, the diversity of warehouse configurations present in modern industry suggests that the adaptability of GRO to different operating environments merits in-depth investigation. Warehouses vary significantly in layout, size, type of stock and degree of automation, each presenting unique challenges for route optimisation. Although our study was based on a warehouse with a specific layout, the GRO algorithm was designed to be adaptable. By modifying the input parameters to reflect the structure of the warehouse, GRO can be adjusted for traditional warehouse layouts, such as those with one or two blocks. For one-block warehouses, where the layout is more straightforward and often linear, the algorithm can focus on optimising the sequence of picking locations within a single block. While in two-block warehouses, it can include optimised transition points to minimise the total distance travelled. Exploring the applicability of GRO in different warehouse configurations will allow us to identify the need for adjustments or modifications to the algorithm to ensure its effectiveness in a broader range of operational scenarios. This research not only increases the practical usefulness of GRO but also contributes to a more comprehensive understanding of how metaheuristics can be adapted to meet the specific needs of different warehouse operations.

This paper contributes to the literature on warehouse logistics by addressing the problem of optimising picking routes in a random storage environment where mixed shelving configurations allow multiple product types in a single storage position. We present a new variant of the FTSP that incorporates two elements: the co-location of multiple products in shared storage bins and prioritising quantity-specific picking requirements. Unlike the original FTSP, which assumes a clear separation between families of nodes, our model dynamically selects picking points from variable storage configurations, ensuring that specific quantities of each product are picked while minimising travel distance.

In managerial terms, the study provides practical insights for improving warehouse operational efficiency by significantly reducing average picking distances. This reduction translates into faster picking times and has indirect benefits related to quality, health, and safety (QHS), improving picker well-being and reducing the likelihood of errors. The implementation of GRO offers warehouse managers a structured route planning framework that reduces the inefficiencies associated with unstructured, intuition-based picking methods. However, this research has limitations, particularly in addressing the dynamic nature of warehouse layout changes and the solution's adaptability over time. While product diversity does not directly impact the model, the ability to quickly adjust to layout modifications and shifting storage requirements remains a challenge. Although the GRO parameters were calibrated for the specific warehouse configuration in this study, further research is needed to develop approaches that can generalize and adapt to larger warehouses or those with frequently changing layouts and operational structures. Other metaheuristics, such as hybrid heuristics, can be investigated in future studies to improve the proposed approach's computational performance.

VII. CONCLUSION

This study represents a development in warehouse logistics optimisation literature, explicitly addressing the underexplored challenge of picking routing in the context of random storage with multiple products per storage location. The complex scenario encountered in a Canadian shoe manufacturer warehouse motivated the development of the GRO algorithm. This approach stands out for its ability to effectively deal with the additional complexity that arises when multiple products are located at the same storage point, a situation common in many modern warehouses but often overlooked in previous research.

Implementing GRO in a warehouse with random stock offers practical and theoretical insights. From a practical point of view, applying GRO promotes more efficient picking operations management, significantly reducing the time and effort required to process orders. Theoretically, our study contributes to the warehouse management literature by introducing a new order-picking problem and offering a solution to picking multiple products in a single position. This research expands current knowledge on route optimisation strategies in complex warehouse environments, highlighting the importance of adapting optimisation approaches to the characteristics of random inventories with multiple products per storage position. In addition, the results obtained with GRO reinforce the applicability of metaheuristics to real logistics problems, demonstrating how algorithmic solutions can be calibrated to address specific operational challenges and significantly improve logistics processes.

Empirical testing on selected product pick lists demonstrated GRO's superiority over existing practices, significantly reducing average pick distances. Statistical analyses, including a paired t-test, substantiated these improvements, illustrating a significant optimisation over the company's prior methods. These findings underscore GRO's potential in diverse operational contexts, suggesting avenues for further research, particularly in adapting GRO to varied warehouse configurations and exploring synergies with other metaheuristics to improve solution generation efficiency.

Throughout this study on the implementation of GRO in a warehouse scenario with random stock, we encountered several noteworthy limitations. Accurately modelling real warehouse conditions posed a significant challenge, especially given the complexity introduced by multiple products in a single position. Additionally, balancing the algorithm's accuracy with computational efficiency required careful calibration of GRO parameters to ensure optimized results without excessive processing time. Collecting and analyzing real warehouse picking data also presented difficulties, particularly ensuring that the dataset captured a broad range of picking scenarios without introducing bias. While these limitations underscore the significant contributions of this study, they also point to crucial areas for further research and refinement in GRO and similar optimisation methods. Future studies could explore the potential of alternative algorithms to address these challenges, potentially uncovering solutions that enhance adaptability and computational efficiency in dynamic and complex warehouse settings.

ACKNOWLEDGMENT

The authors would like to thank the Royer Inc. Team, especially Simon La Rochelle, the company's CEO, for his trust, and David Gaulin, the Head of Software Development, for his technical support, especially with data collection. They would also like to thank the Productique Quebec Team, especially Vincent Thomasset-Laperrière, Director of Research and Partnerships, for their support during the development of this research. This research was supported by

Mitacs through the Mitacs Accelerate program in Canada and by the Natural Sciences and Engineering Research Council (NSERC) of Canada [grant number RGPIN-2024-03880].

AUTHORSHIP CONTRIBUTION STATEMENT

Rodrigo Furlan de Assis: Conceptualization, methodology, investigation, data collection, formal analysis, visualization, writing–original draft, writing–review & editing. William de Paula Ferreira: Conceptualization, methodology, investigation, data collection, formal analysis, visualization, writing–review & editing, resources, validation, funding acquisition, supervision. Alexandre Frias Faria: Investigation, validation, writing–review & editing. Luis Antonio de Santa-Eulalia: Funding acquisition, writing–review & editing. Mustapha Ouhimmou: Funding acquisition, validation, writing–review & editing. Ali Gharbi: Funding acquisition, validation, writing–review & editing.

REFERENCES

[1] R. de Koster, T. Le-Duc, and K. J. Roodbergen, "Design and control of warehouse order picking: A literature review," *Eur. J. Oper. Res.*, vol. 182, no. 2, pp. 481–501, Oct. 2007.

[2] M. Masae, C. H. Glock, and E. H. Grosse, "Order picker routing in warehouses: A systematic literature review," *Int. J. Prod. Econ.*, vol. 224, Jun. 2020, Art. no. 107564.

[3] S. Vanheusden, T. van Gils, K. Ramaekers, T. Cornelissens, and A. Caris, "Practical factors in order picking planning: State-of-the-art classification and review," *Int. J. Prod. Res.*, vol. 61, no. 6, pp. 2032–2056, Mar. 2023.

[4] T. van Gils, K. Ramaekers, A. Caris, and R. B. M. de Koster, "Designing efficient order picking systems by combining planning problems: State-of-the-art classification and review," *Eur. J. Oper. Res.*, vol. 267, no. 1, pp. 1–15, May 2018.

[5] R. De Santis, R. Montanari, G. Vignali, and E. Bottani, "An adapted ant colony optimization algorithm for the minimization of the travel distance of pickers in manual warehouses," *Eur. J. Oper. Res.*, vol. 267, no. 1, pp. 120–137, May 2018.

[6] M. Schiffer, N. Boysen, P. S. Klein, G. Laporte, and M. Pavone, "Optimal picking policies in e-commerce warehouses," *Manage. Sci.*, vol. 68, no. 10, pp. 7497–7517, Oct. 2022.

[7] M.-F. Yang, P.-H. Shih, J. C.-H. Pan, and M.-C. Li, "The optimal layout design for minimizing operating costs in a picker-to-part warehousing system," *Int. J. Adv. Manuf. Technol.*, vol. 118, pp. 2523–2537, Feb. 2022.

[8] N. Boysen, R. de Koster, and F. Weidinger, "Warehousing in the e-commerce era: A survey," *Eur. J. Oper. Res.*, vol. 277, no. 2, pp. 396–411, Sep. 2019.

[9] S. Srinivas and S. Yu, "Collaborative order picking with multiple pickers and robots: Integrated approach for order batching, sequencing and picker-robot routing," *Int. J. Prod. Econ.*, vol. 254, Dec. 2022, Art. no. 108634.

[10] E. Ardjmand, H. Shakeri, M. Singh, and O. S. Bajgiran, "Minimizing order picking makespan with multiple pickers in a wave picking warehouse," *Int. J. Prod. Econ.*, vol. 206, pp. 169–183, Dec. 2018.

[11] D. Battini, M. Calzavara, A. Persona, and F. Sgarbossa, "A method to choose between carton from rack picking or carton from pallet picking," *Comput. Ind. Eng.*, vol. 126, pp. 88–98, Dec. 2018.

[12] J. Leng, D. Yan, Q. Liu, H. Zhang, G. Zhao, L. Wei, D. Zhang, A. Yu, and X. Chen, "Digital twin-driven joint optimisation of packing and storage assignment in large-scale automated high-rise warehouse product-service system," *Int. J. Comput. Integr. Manuf.*, vol. 34, nos. 7–8, pp. 783–800, Aug. 2021.

[13] T. De Lombaert, K. Braekers, R. De Koster, and K. Ramaekers, "In pursuit of humanised order picking planning: Methodological review, literature classification and input from practice," *Int. J. Prod. Res.*, vol. 61, no. 10, pp. 3300–3330, May 2023.

[14] J. P. van der Gaast and F. Weidinger, "A deep learning approach for the selection of an order picking system," *Eur. J. Oper. Res.*, vol. 302, no. 2, pp. 530–543, Oct. 2022.

[15] S. Vanheusden, T. van Gils, A. Caris, K. Ramaekers, and K. Braekers, "Operational workload balancing in manual order picking," *Comput. Ind. Eng.*, vol. 141, Mar. 2020, Art. no. 106269.

[16] R. F. de Assis, A. F. Faria, V. Thomasset-Laperrière, L. A. Santa-Eulalia, M. Ouhimmou, and W. D. P. Ferreira, "Machine learning in warehouse management: A survey," *Proc. Comput. Sci.*, vol. 232, pp. 2790–2799, Jan. 2024.

[17] Ö. Öztürkoğlu and D. Hoser, "A discrete cross aisle design model for order-picking warehouses," *Eur. J. Oper. Res.*, vol. 275, no. 2, pp. 411–430, Jun. 2019.

[18] I. G. Lee, S. H. Chung, and S. W. Yoon, "Two-stage storage assignment to minimize travel time and congestion for warehouse order picking operations," *Comput. Ind. Eng.*, vol. 139, Jan. 2020, Art. no. 106129.

[19] P. Yang, Z. Zhao, and H. Guo, "Order batch picking optimization under different storage scenarios for e-commerce warehouses," *Transp. Res. E, Logistics Transp. Rev.*, vol. 136, Apr. 2020, Art. no. 101897.

[20] N. Shetty, B. Sah, and S. H. Chung, "Route optimization for warehouse order picking operations via vehicle routing and simulation," *Social Netw. Appl. Sci.*, vol. 2, no. 2, pp. 1–18, Feb. 2020.

[21] J. J. R. Reyes, E. L. Solano-Charris, and J. R. Montoya-Torres, "The storage location assignment problem: A literature review," *Int. J. Ind. Eng. Comput.*, vol. 10, no. 2, pp. 199–224, 2019.

[22] X. Xu and C. Ren, "A novel storage location assignment in multi-pickers picker-to-parts systems integrating scattered storage, demand correlation, and routing adjustment," *Comput. Ind. Eng.*, vol. 172, Oct. 2022, Art. no. 108618.

[23] P. Kübler, C. H. Glock, and T. Bauernhansl, "A new iterative method for solving the joint dynamic storage location assignment, order batching and picker routing problem in manual picker-to-parts warehouses," *Comput. Ind. Eng.*, vol. 147, Sep. 2020, Art. no. 106645.

[24] I. Žulj, C. H. Glock, E. H. Grosse, and M. Schneider, "Picker routing and storage-assignment strategies for precedence-constrained order picking," *Comput. Ind. Eng.*, vol. 123, pp. 338–347, Sep. 2018.

[25] D. Loske, M. Klumpp, E. H. Grosse, T. Modica, and C. H. Glock, "Storage systems' impact on order picking time: An empirical economic analysis of flow-rack storage systems," *Int. J. Prod. Econ.*, vol. 261, Jul. 2023, Art. no. 108887.

[26] A. Foroughi, N. Boysen, S. Emde, and M. Schneider, "High-density storage with mobile racks: Picker routing and product location," *J. Oper. Res. Soc.*, vol. 72, no. 3, pp. 535–553, Mar. 2021.

[27] A. Silva, L. C. Coelho, M. Darvish, and J. Renaud, "Integrating storage location and order picking problems in warehouse planning," *Transp. Res. E, Logistics Transp. Rev.*, vol. 140, Aug. 2020, Art. no. 102003.

[28] F. Chen, G. Xu, and Y. Wei, "Heuristic routing methods in multiple-block warehouses with ultra-narrow aisles and access restriction," *Int. J. Prod. Res.*, vol. 57, no. 1, pp. 228–249, Jan. 2019.

[29] X. Guo, Y. Yu, and R. B. M. De Koster, "Impact of required storage space on storage policy performance in a unit-load warehouse," *Int. J. Prod. Res.*, vol. 54, no. 8, pp. 2405–2418, Apr. 2016.

[30] L. F. Morán-Mirabal, J. L. González-Velarde, and M. G. C. Resende, "Randomized heuristics for the family traveling salesperson problem," *Int. Trans. Oper. Res.*, vol. 21, no. 1, pp. 41–57, Jan. 2014.

[31] P. C. Pop, O. Cosma, C. Sabo, and C. P. Sitar, "A comprehensive survey on the generalized traveling salesman problem," *Eur. J. Oper. Res.*, vol. 314, no. 3, pp. 819–835, May 2024.

[32] R. Bernardino and A. Paias, "Solving the family traveling salesman problem," *Eur. J. Oper. Res.*, vol. 267, no. 2, pp. 453–466, 2018.

[33] A. Scholz, S. Henn, M. Stuhlmann, and G. Wäscher, "A new mathematical programming formulation for the single-picker routing problem," *Eur. J. Oper. Res.*, vol. 253, no. 1, pp. 68–84, Aug. 2016.

[34] R. L. Daniels, J. L. Rummel, and R. Schantz, "A model for warehouse order picking," *Eur. J. Oper. Res.*, vol. 105, no. 1, pp. 1–17, Feb. 1998.

[35] G. Casella, A. Volpi, R. Montanari, L. Tebaldi, and E. Bottani, "Trends in order picking: A 2007–2022 review of the literature," *Prod. Manuf. Res.*, vol. 11, no. 1, Dec. 2023, Art. no. 2191115.

[36] W. de Paula Ferreira, F. Armellini, L. A. de Santa-Eulalia, and C. Rebolledo, *Modelling and Simulation in Industry 4.0*. Cham, Switzerland: Springer, 2021, pp. 57–72.

[37] R. F. de Assis, F. M. Guerrini, L. A. Santa-Eulalia, and W. D. P. Ferreira, "An agent-based model for regional market penetration of electric vehicles in Brazil," *J. Cleaner Prod.*, vol. 421, Oct. 2023, Art. no. 138477.

[38] R. F. de Assis, L. A. de Santa-Eulalia, F. Armellini, R. Anholon, I. S. Rampasso, F. M. Guerrini, M. G. Filho, and W. D. P. Ferreira, "A system dynamics approach to unlock the complexity of the S&OP in virtual enterprises," *Enterprise Inf. Syst.*, vol. 17, no. 12, Dec. 2023, Art. no. 2203430.

[39] M. Masae, C. H. Glock, and P. Vichitkunakorn, "A method for efficiently routing order pickers in the leaf warehouse," *Int. J. Prod. Econ.*, vol. 234, Apr. 2021, Art. no. 108069.

[40] M. Reda, A. Onsy, M. A. Elhosseini, A. Y. Haikal, and M. Badawy, "A discrete variant of cuckoo search algorithm to solve the travelling salesman problem and path planning for autonomous trolley inside warehouse," *Knowl.-Based Syst.*, vol. 252, Sep. 2022, Art. no. 109290.

[41] K. Sörensen, "Metaheuristics—The metaphor exposed," *Int. Trans. Oper. Res.*, vol. 22, no. 1, pp. 3–18, Jan. 2015.

[42] R. Montanari, R. Micale, E. Bottani, A. Volpi, and G. La Scalia, "Evaluation of routing policies using an interval-valued TOPSIS approach for the allocation rules," *Comput. Ind. Eng.*, vol. 156, Jun. 2021, Art. no. 107256.

[43] A. Agárdi, L. Kovács, and T. Bányai, "Optimization of automatized picking process," in *Proc. IEEE 13th Int. Symp. Appl. Comput. Intell. Informat. (SACI)*, May 2019, pp. 364–369.

[44] C.-Y. Tsai, J. J. H. Liou, and T.-M. Huang, "Using a multiple-GA method to solve the batch picking problem: Considering travel distance and order due time," *Int. J. Prod. Res.*, vol. 46, no. 22, pp. 6533–6555, Nov. 2008.

[45] G.-H. Wu, C.-Y. Cheng, and M.-H. Liu, "Two-stage metaheuristic algorithms for order-batching and routing problems," *Appl. Sci.*, vol. 12, no. 21, p. 10921, Oct. 2022.

[46] C.-C. Lin, J.-R. Kang, C.-C. Hou, and C.-Y. Cheng, "Joint order batching and picker Manhattan routing problem," *Comput. Ind. Eng.*, vol. 95, pp. 164–174, May 2016.

[47] T.-L. Chen, C.-Y. Cheng, Y.-Y. Chen, and L.-K. Chan, "An efficient hybrid algorithm for integrated order batching, sequencing and routing problem," *Int. J. Prod. Econ.*, vol. 159, pp. 158–167, Jan. 2015.

[48] F. Chen, G. Xu, and Y. Wei, "An integrated metaheuristic routing method for multiple-block warehouses with ultranarrow aisles and access restriction," *Complexity*, vol. 2019, no. 1, Jan. 2019, Art. no. 1280285.

[49] E. G. Pardo, S. Gil-Borrás, A. Alonso-Ayuso, and A. Duarte, "Order batching problems: Taxonomy and literature review," *Eur. J. Oper. Res.*, vol. 313, no. 1, pp. 1–24, Feb. 2024.

[50] J. I. U. Rubrico, J. Ota, T. Higashi, and H. Tamura, "Metaheuristic scheduling of multiple picking agents for warehouse management," *Ind. Robot, Int. J.*, vol. 35, no. 1, pp. 58–68, Jan. 2008.

[51] F. Chen, H. Wang, C. Qi, and Y. Xie, "An ant colony optimization routing algorithm for two order pickers with congestion consideration," *Comput. Ind. Eng.*, vol. 66, no. 1, pp. 77–85, Sep. 2013.

[52] P. Cortés, R. A. Gómez-Montoya, J. Muñuzuri, and A. Correa-Espinal, "A Tabu search approach to solving the picking routing problem for large- and medium-size distribution centres considering the availability of inventory and K heterogeneous material handling equipment," *Appl. Soft Comput.*, vol. 53, pp. 61–73, Apr. 2017.

[53] A. H. Schrotenboer, S. Wruck, K. J. Roodbergen, M. Veenstra, and A. S. Dijkstra, "Order picker routing with product returns and interaction delays," *Int. J. Prod. Res.*, vol. 55, no. 21, pp. 6394–6406, Nov. 2017.

[54] T. Chabot, R. Lahyani, L. C. Coelho, and J. Renaud, "Order picking problems under weight, fragility and category constraints," *Int. J. Prod. Res.*, vol. 55, no. 21, pp. 6361–6379, Nov. 2017.

[55] J. Li, R. Huang, and J. B. Dai, "Joint optimisation of order batching and picker routing in the online retailer's warehouse in China," *Int. J. Prod. Res.*, vol. 55, no. 2, pp. 447–461, Jan. 2017.

[56] T. Bódis and J. Botzheim, "Bacterial memetic algorithms for order picking routing problem with loading constraints," *Exp. Syst. Appl.*, vol. 105, pp. 196–220, Sep. 2018.

[57] F. Weidinger, "Picker routing in rectangular mixed shelves warehouses," *Comput. Oper. Res.*, vol. 95, pp. 139–150, Jul. 2018.

[58] E. Ardjmand, O. S. Bajgiran, and E. Youssef, "Using list-based simulated annealing and genetic algorithm for order batching and picker routing in put wall based picking systems," *Appl. Soft Comput.*, vol. 75, pp. 106–119, Feb. 2019.

[59] T. van Gils, A. Caris, K. Ramaekers, and K. Braekers, "Formulating and solving the integrated batching, routing, and picker scheduling problem in a real-life spare parts warehouse," *Eur. J. Oper. Res.*, vol. 277, no. 3, pp. 814–830, Sep. 2019.

[60] E. Bottani, G. Casella, and T. Murino, "A hybrid metaheuristic routing algorithm for low-level picker-to-part systems," *Comput. Ind. Eng.*, vol. 160, Oct. 2021, Art. no. 107540.

[61] S. Gil-Borrás, E. G. Pardo, A. Alonso-Ayuso, and A. Duarte, "A heuristic approach for the online order batching problem with multiple pickers," *Comput. Ind. Eng.*, vol. 160, Oct. 2021, Art. no. 107517.

[62] Z. Düzgit, A. Ö. Toy, and A. C. Saner, "Performance comparison of metaheuristics for the multiblockwarehouse order picking problem," *Int. J. Ind. Eng.*, vol. 28, no. 1, pp. 75–91, 2021.

[63] Ç. Cergibozan and A. S. Tasan, "Genetic algorithm based approaches to solve the order batching problem and a case study in a distribution center," *J. Intell. Manuf.*, vol. 33, no. 1, pp. 137–149, Jan. 2022.

[64] J. A. Cano, P. Cortés, J. Muñuzuri, and A. Correa-Espinal, "Solving the picker routing problem in multi-block high-level storage systems using metaheuristics," *Flexible Services Manuf. J.*, vol. 35, no. 2, pp. 376–415, Jun. 2023.

[65] R. Wu, J. He, X. Li, and Z. Chen, "A memetic algorithm with fuzzy-based population control for the joint order batching and picker routing problem," *Inf. Sci.*, vol. 656, Jan. 2024, Art. no. 119913.

[66] D. Battini, M. Calzavara, A. Persona, and F. Sgarbossa, "Additional effort estimation due to ergonomic conditions in order picking systems," *Int. J. Prod. Res.*, vol. 55, no. 10, pp. 2764–2774, May 2017.

[67] E. H. Grosse, C. H. Glock, and W. P. Neumann, "Human factors in order picking: A content analysis of the literature," *Int. J. Prod. Res.*, vol. 55, no. 5, pp. 1260–1276, Mar. 2017.

[68] H. M. Gámez Albán, T. Cornelissens, and K. Sörensen, "A new policy for scattered storage assignment to minimize picking travel distances," *Eur. J. Oper. Res.*, vol. 315, no. 3, pp. 1006–1020, Jun. 2024.

[69] A. R. F. Pinto and M. S. Nagano, "Genetic algorithms applied to integration and optimization of billing and picking processes," *J. Intell. Manuf.*, vol. 31, no. 3, pp. 641–659, Mar. 2020.

[70] L. Pansart, N. Catusse, and H. Cambazard, "Exact algorithms for the order picking problem," *Comput. Oper. Res.*, vol. 100, pp. 117–127, Dec. 2018.

[71] S. K. Iyer and B. Saxena, "Improved genetic algorithm for the permutation flowshop scheduling problem," *Comput. Oper. Res.*, vol. 31, no. 4, pp. 593–606, Apr. 2004.

[72] O. Kramer and O. Kramer, *Genetic Algorithms*. Berlin, Germany: Springer, 2017.

[73] S. Mirjalili, *Genetic Algorithm*. Cham, Switzerland: Springer, 2019, pp. 43–55.

[74] A. P. Engelbrecht, *Computational Intelligence: An Introduction*. Hoboken, NJ, USA: Wiley, 2007.

RODRIGO FURLAN DE ASSIS received the D.Sc. degree. He is currently a Postdoctoral Fellow with École de Technologie Supérieure (ÉTS), Montreal, Canada, with a focus on modeling and simulation of manufacturing systems and integrating artificial intelligence in decision-making processes for Industry 4.0. He is the author of several journal articles and conference papers, with research interests in operational research, optimization, and production systems.

WILLIAM DE PAULA FERREIRA received the Ph.D. degree. He is currently an Associate Professor with the Department of Systems Engineering, École de Technologie Supérieure (ÉTS). Before his professorship, he worked for several years as a practitioner in industrial engineering, mainly in the automotive and petrochemical sectors. He has co-authored more than 40 scientific publications in peer-reviewed journals, conference proceedings, and book chapters. His research interests include Industry 4.0, intelligent manufacturing, modeling and simulation, and digital twins.

ALEXANDRE FRIAS FARIA received the D.Sc. degree. He is currently a Mathematician who has been combining ideas from mathematics and computer science to solve combinatorial optimization problems by developing efficient algorithms. His work in industry and academia is mainly related to routing and partition problems. His current research interests include mathematical programming and related exact methods.

LUIS ANTONIO SANTA-EULALIA received the Ph.D. degree. He is currently a Full Professor in operations management with the École de gestion, Université de Sherbrooke, Canada. With a background in industrial engineering, he is a Researcher with the Research Center on Intelligent Organizations, as well as a member of diverse research groups and networks, such as the Sustainable, Digital and Green (SDG) Innovation Network in Canada. He has co-authored more than 180 articles published in peer-reviewed journals or presented at conferences with selective editorial policies. His current research interests include emergent technologies, novel business models, and practices for innovative and sustainable operations and supply chain management.

MUSTAPHA OUHIMMOU received the Ph.D. degree. He is currently a Full Professor with the Department of Systems Engineering, École de Technologie Supérieure (ÉTS). He is also a Researcher at the Centre interdisciplinaire de recherche en opérationnalisation du développement durable, with expertise in sustainable logistics, urban and reverse logistics, and intelligent transport systems. His work focuses on optimizing supply chains through vehicle routing optimization, automated warehousing and distribution, and inventory management. He also applies Industry 4.0 technologies, including digital twins and artificial intelligence, to enhance forestry, mining, healthcare, e-commerce, and construction logistics processes.

ALI GHARBI received the Ph.D. degree. He is currently a Full Professor with the Department of Systems Engineering, École de Technologie Supérieure (ÉTS). He is a leading expert in industrial engineering with significant contributions to optimizing unreliable manufacturing systems. His work has advanced the field by developing stochastic feedback control policies for manufacturing and remanufacturing systems, particularly in dynamic and unreliable environments. Over the past 30 years, he and his team have developed and refined a methodology based on the integration of optimal stochastic control theory with simulation-based optimization models. This methodology has made it possible to solve many complex problems related to the integration of production and maintenance strategies to handle machine failures, random demand variations, and product quality and perishability issues. His research has been widely recognized, earning over 6000 citations and an H-index of 46. He has also supervised numerous postdoctoral researchers and Ph.D. and master's students, many of whom now hold prestigious academic and industrial positions. His collaboration with scholars and industries across various domains has resulted in over 130 high-impact publications in top-tier journals. He continues to innovate by developing policies that balance economic and environmental concerns.