Sri Lanka Society of Transport and Logistics

**Journal of South Asian Logistics and Transport**
Volume 5 Issue 1 March 2025
JSALT 5.1.4: pp. 69-85
DOI: https://doi.org/10.4038/jsalt.v5i1.101

# A FRAMEWORK FOR SAFE AND EFFICIENT FISHBONE LAYOUTS IN WAREHOUSE OPERATIONS

***S Nimishan*** [a*] ***and K Theekshani*** [b]

[a] *Centre for Computer Studies, Sabaragamuwa University of Sri Lanka, Sri Lanka*
[b] *Department of Maritime Transportation and Logistics, Ocean University of Sri Lanka, Sri Lanka*

* Correspondence should be addressed to *nimishan@ccs.sab.ac.lk*

---

**ABSTRACT**

*The adoption of Industry 4.0 technologies has transformed warehouse management, enhancing its efficiency, accuracy, and profitability. However, optimising warehouse layouts, particularly fishbone designs, remains complex and often relies on intricate mathematical models that are impractical for workers to implement. Addressing this gap, this study proposes a straightforward methodology for generating optimised fishbone layouts adaptable to various warehouse sizes. The framework integrates warehouse, product, and forklift dimensions, leveraging Prolog to calculate aisle widths, row numbers, and forklift turning capabilities. This dynamic approach optimises safety and operational efficiency while simplifying decision-making processes. Validation through real-world scenarios demonstrates the model's adaptability and effectiveness across diverse setups, ensuring safe forklift navigation and efficient space utilization. By bridging the divide between theoretical models and practical application, this methodology offers warehouse operators an accessible tool for layout design. The findings contribute to warehouse optimization research, providing a scalable and efficient solution for optimising space and operations in warehouse management systems.*

**Keywords:** *Aisle configuration, automation, fishbone layout, prolog, warehouse design*

## 1. INTRODUCTION

A warehouse is a specialised facility designed to store goods for commercial purposes, serving as a critical hub within the supply chain. These structures play a vital role in various industries, particularly in the food supply chain, where they ensure the safe and efficient storage of perishable and non-perishable goods [1]. Warehouses are used by a diverse range of stakeholders, including manufacturers, importers, exporters, wholesalers, and transport companies, each relying on the warehouses' capacity to manage inventory and facilitate the smooth movement of goods. The operational efficiency of these warehouses is integral to the overall performance of the logistics industry, as their ability to streamline storage, handling, and distribution processes directly influences the effectiveness and reliability of the entire supply chain [2], [3]. Therefore, the design and management of warehouse operations are key factors in optimising logistics efficiency and supporting the seamless flow of goods from producers to consumers.

With the advent of Industry 4.0 in 2011, the concept of the smart warehouse, also known as Warehousing 4.0, has experienced significant growth. Industry 4.0 marks the fourth industrial revolution and is widely recognized as one of the most efficient and effective strategies for increasing return on investment (ROI) [4]. By minimising the need for manual labour, enhancing precision, and optimising operational processes, Industry 4.0 technologies contribute to significant improvements in overall warehouse performance [5]. The integration of advanced technologies into warehouse operations is reshaping the industry, driving greater efficiency, accuracy, and profitability.

A typical warehouse often features a conventional layout with parallel or vertical aisles and shelves as illustrated in Figure 1. These traditional designs, which rely heavily on manual processes, are prone to several inefficiencies that hinder optimal performance [6]. Issues such as poor space utilization limit storage capacity and complicate access, while inefficient material handling requires longer travel distances for workers and equipment, increasing internal movement time. The rigidity of traditional layouts further complicates adaptation to changing inventory levels and operational demands, slowing operations and reducing responsiveness. Additionally, scalability is challenging, with costly and difficult expansions or reconfigurations [7]. These inefficiencies, particularly in picking operations, highlight the need for improvements in reducing picking distances and enhancing work efficiency, especially in developing countries [8].

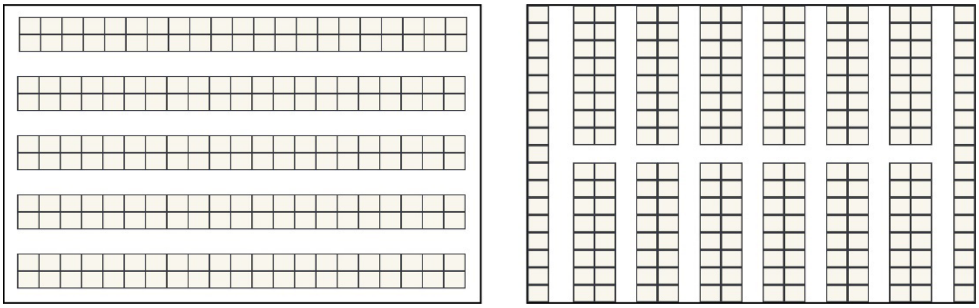

**Figure 1: Two common types of traditional warehouse layouts**

Similar to other sectors, industrial warehouses have undergone significant evolution due to advancements in science and technology [5]. Even during the first and second industrial revolutions, the role of warehouses began to change, becoming more specialised to meet emerging needs. The mass production of goods during these periods spurred the development of larger, more specialised warehouses, strategically situated near transport hubs to facilitate the efficient movement and storage of products [9].

Given these challenges, it becomes essential to organize product storage within the warehouse strategically. In 2009, Gue and Meller introduced the fishbone layout with a single Pick-up and Deposit point, which can be divided into four distinct zones based on the placement of diagonal, horizontal, and vertical cross aisles [10]. They also found that, when implemented in a single-instruction unit load warehouse, the fishbone layout reduced the expected travel distance by 20% compared to traditional layouts. It is important to note, however, that while the fishbone layout offers operational benefits, The warehouse layout construction cost functions presented in existing literature are often highly complex and require extensive computation [11]. These functions involve complex mathematical models and calculations, which can be difficult for warehouse workers to understand and implement [12]. Such complexity makes them impractical for daily use by those directly involved in warehouse operations, as they demand specialised knowledge or software that are not typically accessible to workers on the ground.

While existing research explores various layout optimization methods, there is a noticeable gap in studies that focus on adaptable approaches for both small-scale and large-scale warehouses. Specifically, most studies fail to address the challenges of designing fishbone layouts in a way that is computationally feasible and accessible for real-world applications. This gap underscores the need for further research into methods that can effectively optimize fishbone layouts while considering resource limitations and warehouse size variability.

This study aims to design a simple model for a warehouse using a dedicated storage method. This approach offers solutions for optimising the use of available warehouse space by efficiently organizing product placement. To support this model, an algorithm is implemented that calculates the number of rows within the warehouse under a fishbone layout. The program also determines the distribution of rows across the different pick zones ($S_1$, $S_2$, $S_3$ and $S_4$) created by the diagonal aisles, as shown in Figure 2.

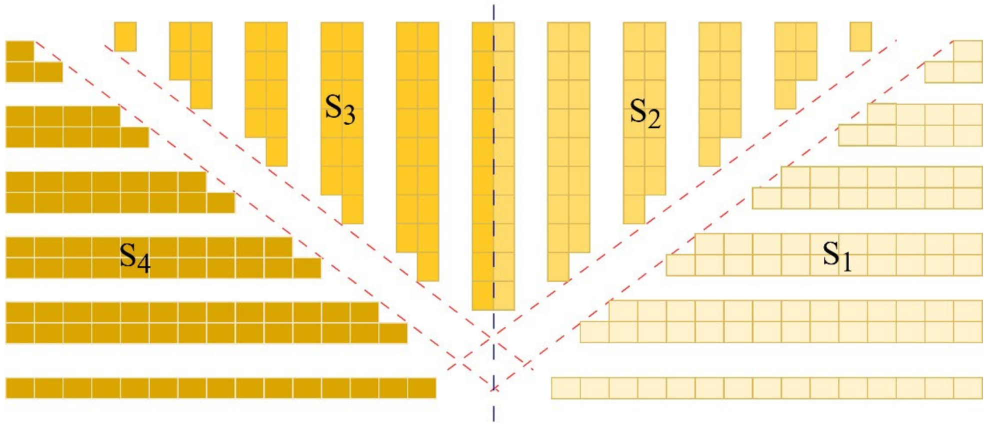


**Figure 2: Example of a fishbone layout with designated pick zones $S_1$, $S_2$, $S_3$, and $S_4$**

The core calculations are as follows:

- The total number of rows is determined by dividing the warehouse width by the width of the picking aisle.
- The total number of columns is found by dividing half the warehouse length by the width of the double-row shelves.
- The rows are then evenly distributed among the four pick zones, ensuring that each zone has an optimal number of rows for efficient picking operations.

By integrating these calculations into the simulation model, the warehouse layout can be optimised to maximize space utilization and enhance the efficiency of product retrieval. This approach is designed to improve overall operational effectiveness within the warehouse.

In addition to implementing a fishbone storage layout, optimising the picking path is also crucial for enhancing warehouse productivity which reduces material retrieval times and shortens travel distances within the facility. An efficient picking path, achieved through the fishbone layout with diagonal aisles, minimizes travel time and reduces labour costs by providing multiple access points to each pick zone and eliminating backtracking. This strategic organisation of products and alignment of picking paths with the warehouse design streamline the picking process, leading to faster order fulfilment and increased productivity. Consequently, this study will also offer insights into the precise area requirements for finished goods warehouses, addressing key aspects of warehouse efficiency and modern supply chain demands.

## 2. LITERATURE REVIEW

### 2.1. Industry 4.0 Integration into Fishbone Warehousing Design

In recent years, the rapid growth of e-commerce has led to an increased demand for faster and more efficient order-picking processes. Researchers have extensively studied warehouse layouts and the optimization of picking paths to address this challenge [13], [14], [15]. However, research on the fishbone layout is relatively rare and has only emerged in recent years. In 2009, Gue and Meller deviated from traditional approaches and introduced two innovative warehouse internal layouts featuring inclined cross-aisles including the V-shaped and fishbone layouts [10].

With the advancement of Industry 4.0 technologies, recent studies have explored integrating automation and optimization techniques into fishbone layouts. Kumar and Kumar [16] found that mobile robots have a greater number of feasible paths in fishbone layouts compared to traditional warehouse designs, reducing collisions in order-picking systems. Similarly, Wanying et al. [17] developed an improved fishbone layout incorporating an embedded central corridor to optimize Automated Guided Vehicle (AGV) routes, reducing material handling distances by 12.8%. Wan and Liu. [18] further extended this approach by developing a 3D design optimization model for fishbone layouts in brownfield warehouses, minimising average travel distances. These studies indicate the potential benefits of Industry 4.0-driven optimisations for fishbone layouts, yet gaps remain in adapting these solutions to computationally constrained environments and real-world warehouse conditions.

### 2.2. Practical Challenges in Fishbone Layouts

While fishbone layouts offer efficiency improvements, their implementation presents several practical challenges. Pohl et al. [19] discovered that the fishbone layout could reduce expected travel distance by 10-15% in unit load warehouses using dual command operations. This finding highlights its efficiency benefits; however, the feasibility of implementing fishbone layouts across different warehouse configurations remains a challenge.

Cardona et al. [20] developed a nonlinear optimisation model to determine the optimal slope of diagonal cross aisles, minimising expected travel time. However, variations in rack geometries and aisle sizes across industries limit the direct application of such models. Similarly, Clark and Meller [21] accounted for forklift travel in fishbone layouts, concluding that optimal shapes remain valid regardless of rack height. Despite these findings, practical adoption remains limited due to the complexity of implementation in diverse warehouse environments.

A study using MATLAB-based simulations analysed a stochastic model for return-type picking paths in fishbone and V-shaped layouts. The authors developed a storage area utilisation model, selecting a curved main channel and validating the approach through MATLAB simulations [22]. Additionally, Zhou et al. [23] employed metaheuristic algorithms, including genetic algorithms, ant colony optimisation, and the cuckoo algorithm, to optimise picking paths in fishbone layouts. Their approach demonstrated improvements in efficiency, yet the adaptability of such models in real-world warehouses with resource constraints is unclear. The authors in [24] further explored storage location assignment using the firework algorithm, showing improvements in order-picking efficiency. However, these models often assume idealised conditions, overlooking real-world factors such as computational limitations and worker adaptability.

Although path optimisation and efficient route finding in warehouse management are well-established areas of research, recent studies have increasingly focused on the fishbone layout as a promising solution for improving picking efficiency, as shown in the timeline of recent works in Fishbone layout in Figure 3. However, many of these studies overlook the practical challenges of calculating and implementing fishbone layouts in computational resource-constrained environments, as well as the warehouse workers' computer literacy, which can further complicate the adoption and execution of these advanced layouts. Specifically, existing research often fails to explore methods of arranging fishbone layouts in a way that is adaptable to both small-scale and large-scale warehouses. Additionally, challenges such as computational constraints, workforce digital literacy, and real-world adaptability remain underexplored, limiting the wider adoption of fishbone layouts in logistics operations. Addressing these gaps is crucial for developing scalable and efficient warehouse design solutions that can accommodate varying resource constraints while enhancing overall operational efficiency.

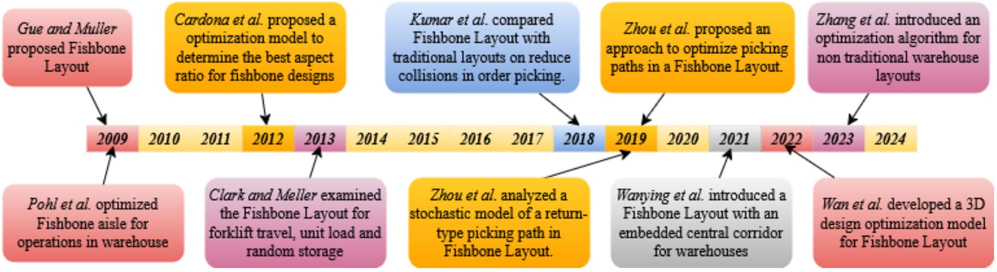


**Figure 3: Timeline of recent works in fishbone layout-based warehouse management**

## 3. METHODOLOGY

This study presents a methodology and simulation model for determining the optimal fishbone layout of a warehouse storage system, integrating the dimensions of the warehouse, products, and forklifts. The approach aims to develop an optimised warehouse layout by considering the interaction between these factors to ensure an efficient arrangement of products while maintaining safe forklift navigation. Using Prolog, a powerful logic-based programming language, the problem is modelled in a flexible and dynamic way [23], [24]. Prolog's declarative nature enables the representation of various constraints—such as aisle width, number of rows, and forklift turning space—while automatically computing the optimal layout. The system iteratively adjusts the number of rows in both horizontal and vertical directions, ensuring that forklifts can safely manoeuvre through the aisles without colliding with shelves. Additionally, another crucial aspect of this process is calculating the forklift's turning ability, which requires determining the turning angle $\theta$ in degrees [18]. This angle is computed based on the warehouse width and length, ensuring that forklifts have enough space to turn without hitting the shelves. This methodology facilitates the design of an effective fishbone storage layout while providing a systematic approach to dynamically adjust the layout in response to different warehouse and product parameters. This article presents the optimised design for the fishbone layout, illustrated in Figure 4.

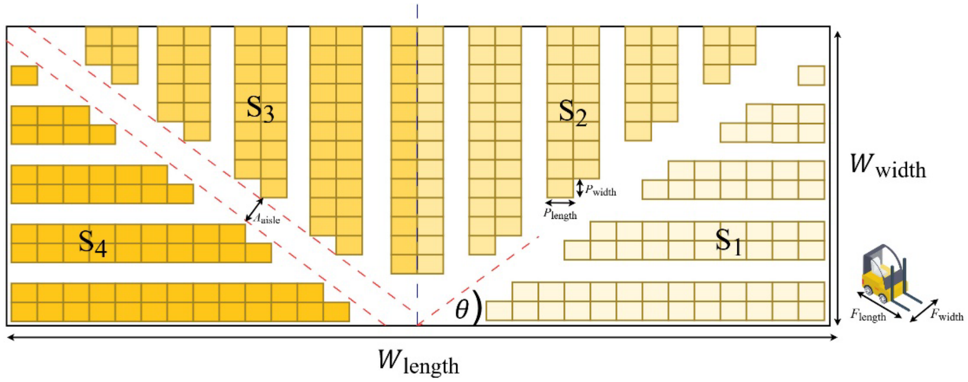


**Figure 4: Diagram of the fishbone layout warehouse with the utilised parameters**

The parameters of this article are set as follows:

- Warehouse width: $W_{width}$
- Warehouse length: $W_{length}$
- Product width: $P_{width}$
- Product length: $P_{length}$
- Forklift width: $F_{width}$
- Forklift length: $F_{length}$

In addition, some assumptions about the environment setup of the warehouse need to be made before the model is established. The specific assumptions are as follows:

- The warehouse has a rectangular shape with fixed width and length.
- The forklift must be able to safely navigate between the shelves without colliding with them.
- The aisle width is derived from the forklift's dimensions to ensure safe turning space.
- The height of the warehouse or the shelves is not considered in this model.

The fishbone layout is designed by determining the number of horizontal rows $S_1$ and vertical rows $S_2$ that can be placed in the warehouse while ensuring forklift safety. $S_3$ and $S_4$ are derived from $S_1$ and $S_2$, respectively, due to the symmetry of the layout. The aisle width is the critical factor for determining if the forklift can turn and navigate between rows without collision. The aisle width $A_{aisle}$ is calculated as follows:

$$A_{aisle} = \max(2 \cdot F_{width}, 1.5 \cdot F_{length}) \text{ -----------------------------------(1)}$$

Here, the aisle width is chosen based on two criteria: it is either twice the forklift width to ensure there is enough space for smooth navigation, or it is 1.5 times the forklift length to allow for safe turning without hitting the shelves. The larger of these two values is used to determine the aisle width [27]. The number of horizontal rows $S_1$ is determined by the available warehouse width $W_{width}$ and the combined space required for each row, which includes both the product width $P_{width}$ and the aisle width $A_{aisle}$. The formula for $S_1$ is:

$$S_1 = \left\lfloor \frac{W_{width}}{P_{width} + A_{aisle}} \right\rfloor \text{ ----------------------------------------------------------(2)}$$

Here, $\lfloor x \rfloor$ denotes the floor function, which rounds down to the nearest integer. This ensures that only full rows that fit into the available space are counted. Similarly, the number of vertical rows $S_2$ is determined by the available warehouse length $W_{length}$ and the combined space required for each row, which includes both the product length $P_{length}$ and the aisle width $A_{aisle}$. The formula for $S_2$ is:

$$S_2 = \left\lfloor \frac{W_{length}}{P_{length} + A_{aisle}} \right\rfloor \text{ ---------------------------------------------------------(3)}$$

Again, the floor function ensures that the rows fit completely within the warehouse length. The forklift's turning angle is calculated based on the warehouse dimensions. The turning angle $\theta$ is determined using the warehouse width and length as follows:

$$\theta = \arctan\left(\frac{W_{width}}{W_{length}/2}\right) \text{ -----------------------------------------------------(4)}$$

Where $\theta$ is the turning angle in radians. This can be converted to degrees by multiplying the result by $\frac{180}{\pi}$, therefore:

$$\theta_{degrees} = \arctan\left(\frac{W_{width}}{W_{length}/2}\right) \cdot \frac{180}{\pi} \text{ ---------------------------------------------(5)}$$

This calculated angle ensures that the forklift has enough space to turn at the aisle intersections without colliding with the shelves.

To ensure that the forklift can safely navigate between rows, it is necessary to check whether there is enough space for the forklift to turn without hitting the shelves. This safety condition is based on the free space available after accounting for the shelf depth. The formula for checking safety is:

$$\text{Free Space} = W_{\text{width}} - 2 \cdot P_{\text{length}} \text{ ---------------------------------------------(6)}$$

This free space must be greater than or equal to the aisle width $A_{aisle}$ for the forklift to turn safely. The safety conditions are therefore:

$$\text{Safety} = \begin{cases} \text{yes} & \text{if} \quad \text{FreeSpace} \geq A_{\text{aisle}} \\ \text{no} & \text{if} \quad \text{FreeSpace} < A_{\text{aisle}} \end{cases} \text{------------------------------------------}(7)$$

If the safety condition is not satisfied, the model will adjust the number of rows until a safe configuration is found.

The implementation uses Prolog to compute the layout dynamically. Given the warehouse and product dimensions, the system first calculates the aisle width, then the number of rows that can be fit both horizontally and vertically. The forklift's ability to turn is checked, and if necessary, the model adjusts the number of rows to ensure safety.

**Table 1: Setup fishbone layout algorithm**

Require: $W_{Width}, W_{Length}, P_{Width}, P_{Length}, F_{Width}, F_{Length}$
Ensure: $RowsS1, RowsS2, Theta\ Safe$

Compute Theta:

$Theta \leftarrow \text{atan}\left(W_{Width}/\left(W_{Length}/2\right)\right) \cdot (180/\pi)$

$Aisle_{Width} \leftarrow \max\left(2 \cdot F_{Width}, 1.5 \cdot F_{Length}\right)$

Calculate Initial Rows:

$RowsS1 \leftarrow \lfloor W_{Width} \,/\, (P_{Width} + Aisle_{Width}) \rfloor$

$RowsS2 \leftarrow \lfloor W_{Length} \,/\, P_{Length} + Aisle_{Width}) \rfloor$

$Safe \leftarrow \left(W_{Width} - 2 \cdot P_{Length}\right) \geq Aisle_{Width}$

Calculate Initial Rows:

$RowsS1 \leftarrow \lfloor W_{Width} \,/\, (P_{Width} + Aisle_{Width}) \rfloor$

$RowsS2 \leftarrow \lfloor W_{Length} \,/\, (P_{Length} + Aisle_{Width}) \rfloor$

$Safe \leftarrow \left(W_{Width} - 2 \cdot P_{Length}\right) \geq Aisle_{Width}$

$While\ Safe = false\ do$

$If\ RowS1 > 1\ then$

$RowsS1 \leftarrow RowsS1 - 1$

$else$

$RowsS1 \leftarrow RowsS1 + 1$

$end\ if$

$Recalculate\ Safe$

$end\ while$

$Return$: $RowsS1, RowsS2, Theta, Safe$

The general steps are as follows:

- **Calculate the aisle width** based on forklift dimensions.

- **Compute the initial number of rows** for both horizontal and vertical placement.
- **Calculate the forklift's turning angle** $\boldsymbol{\theta}$ using the warehouse width and length, ensuring the forklift has enough space to turn safely between rows.
- **Check if the forklift can safely navigate** through the aisles.
- **Adjust the number of rows dynamically** if necessary, until a safe layout is achieved.

This methodology allows for the systematic and safe design of a fishbone storage layout. Algorithm 1 is designed to calculate the number of rows based on the available space, product dimensions, and forklift specifications while ensuring forklift manoeuvrability. The model determines an optimal layout that maximises storage capacity while maintaining operational safety. The iterative adjustment of row numbers ensures that a valid and safe layout is always achieved.

The implementation of the model was carried out using Prolog, specifically using SWI-Prolog. SWI-Prolog development has been driven by the needs of real-world applications, offering a powerful environment for dynamic problem-solving and logical programming. The Prolog-based approach allowed for the efficient calculation and adjustment of warehouse layouts, ensuring a smooth integration of the optimisation algorithm with the dynamic adjustments of rows and aisle width for safety. Additionally, various queries were executed on different warehouse, product, and forklift sizes to test and validate the model's adaptability and effectiveness in diverse real-world scenarios.

## 4. ANALYSIS AND DISCUSSION

This study classifies building sizes into three categories based on their total square footage: small (<5000 sq. ft.), medium (5000–10,000 sq. ft.), and large (>10,000 sq. ft.). This categorisation allowed for a more nuanced analysis of the data, enabling comparisons across diverse building scales and providing insights into how size influences various metrics under investigation. Establishing clear thresholds ensured consistency in evaluations, facilitating robust interpretations and meaningful comparisons across the small, medium, and large warehouse categories. The analysis incorporated varying product dimensions and forklift dimensions, as explored through Prolog queries.

### 4.1. Numerical Example

This section illustrates the use of the method—through a hypothetical numerical example—and the insights gained. A warehouse with dimensions of 60 feet in width and 80 feet in length is considered, along with products and forklifts, both measuring

3 feet in width and 3 feet in length for the products, and 4 feet in length for the forklift. The Prolog queries first calculated the angle (Theta) for the forklift's turning manoeuvre. Based on the warehouse width (60 feet) and length (80 feet), the calculation revealed that the turning angle was approximately 56.3 degrees, which is derived from the ratio of the warehouse width to half the warehouse length. The algorithm then calculates the required aisle width, which is determined by the forklift's dimensions. It uses the larger of two values: twice the forklift's width or 1.5 times the forklift's length. The larger value is chosen, so the aisle width is 4.5 meters.

Next, the method determined the number of horizontal rows (zone S1) and vertical rows (zone S2) that could be accommodated. For S1, the width of the warehouse (60 feet) was divided by the combined width of the product (3 feet) and the aisle width, which was calculated using the forklift dimensions. The result was 6 horizontal rows. Similarly, for S2, the length of the warehouse (80 feet) was divided by the combined product length (3 feet) and aisle width, yielding 8 vertical rows. Finally, the safety check confirmed that the forklift could safely navigate the layout, with no collisions, leading to the conclusion that the configuration is safe for operation. These calculations illustrate how the method provides critical insights for warehouse layout optimisation, balancing both space utilisation and safety requirements.

**Table 2: Simulation results for small warehouse sizes considered in this article**

| Width | Length | $P_{Width}$ | $P_{Length}$ | $F_{Width}$ | $F_{Length}$ | S1 | S2 | Theta | $A_{Width}$ |
|---|---|---|---|---|---|---|---|---|---|
| 5 | 30 | 1 | 1 | 1 | 2 | 3 | 7 | 45 | 3 |
| 20 | 35 | 1 | 1 | 2 | 3 | 3 | 6 | 48.81 | 4.5 |
| 20 | 30 | 2 | 2 | 1 | 2 | 3 | 5 | 53.13 | 3 |
| 30 | 40 | 2 | 2 | 2 | 3 | 4 | 6 | 56.31 | 4.5 |
| 25 | 45 | 3 | 3 | 2 | 3 | 2 | 5 | 48.01 | 4.5 |

**Table 3: Simulation results for medium warehouse sizes considered in this article**

| Width | Length | $P_{Width}$ | $P_{Length}$ | $F_{Width}$ | $F_{Length}$ | S1 | S2 | Theta | $A_{Width}$ |
|---|---|---|---|---|---|---|---|---|---|
| 50 | 70 | 1 | 1 | 2 | 3 | 9 | 12 | 55.01 | 4.5 |
| 60 | 80 | 1 | 1 | 2 | 3 | 10 | 14 | 56.31 | 4.5 |
| 65 | 85 | 2 | 2 | 3 | 4 | 8 | 10 | 56.82 | 6 |
| 75 | 90 | 4 | 2 | 2 | 4 | 6 | 11 | 59.04 | 6 |
| 100 | 90 | 4 | 4 | 3 | 4 | 8 | 7 | 65.77 | 6 |

This example demonstrates the method's ability to compute warehouse configurations effectively, ensuring both operational efficiency and safety in the design of warehouse layouts.

The results of these evaluations are detailed in Table 1, which presents data on small warehouses and other dimensions of product and forklift; Table 2, which outlines the findings for medium warehouses and other dimensions of product and forklift; and Table 3, which focuses on large warehouses with other dimensions of product and forklift, providing a comprehensive view of the relationship between warehouse size and operational parameters. These classifications also align with existing literature and industry standards, enhancing the relevance and applicability of our findings.

**Table 4: Simulation results for large warehouse sizes considered in this article**

| Width | Length | $P_{Width}$ | $P_{Length}$ | $F_{Width}$ | $F_{Length}$ | S1 | S2 | Theta | $A_{Width}$ |
|---|---|---|---|---|---|---|---|---|---|
| 100 | 150 | 5 | 4 | 3 | 4 | 6 | 12 | 53.13 | 6 |
| 150 | 160 | 5 | 4 | 4 | 3 | 10 | 13 | 61.93 | 8 |
| 150 | 180 | 5 | 4 | 4 | 3 | 10 | 15 | 59.04 | 8 |
| 150 | 180 | 5 | 4 | 2 | 3 | 10 | 15 | 59.04 | 4.5 |
| 220 | 180 | 6 | 6 | 3 | 3 | 12 | 10 | 67.75 | 6 |

### 4.2. Goodness of Solution

The computational experiments were performed on a computer equipped with an Intel(R) i5-7300U CPU @ 2.60GHz processor and 8GB of RAM. During the execution of SWI-Prolog, the system used approximately 16.5 MB of memory and 0.3% of CPU resources, demonstrating efficient resource usage.

The proposed methodology effectively integrates Prolog's logic-based computation with warehouse layout optimisation. By iterating through potential configurations, adjusting the number of rows dynamically, and verifying safety conditions, the system guarantees that the layout remains both space-efficient and forklift-safe. The use of Prolog's backtracking and logical reasoning features ensures that the solution is adaptable and can accommodate various constraints. Ultimately, this methodology provides a robust and flexible approach to warehouse layout design, ensuring that all relevant factors are considered and that an optimal solution is achieved for any given set of warehouse dimensions, product sizes, and forklift specifications.

### 4.3. Limitations of the Research

While the proposed methodology is adaptable to various warehouse sizes, it may not support highly specialised or unconventional warehouse environments. Future research could expand the framework to handle even more diverse warehouse types or consider additional parameters. Also, the current model assumes ideal conditions for factors such as rectangular warehouse shapes, forklift turning capabilities and aisle widths. However, real-world constraints such as non-rectangular configurations, fluctuating product dimensions, unforeseen operational disruptions, or irregular warehouse layouts may affect the outcomes.

## 5. CONCLUSIONS

This research presents an innovative approach to generating optimised fishbone layouts in warehouse management, addressing the challenges of complexity and practicality that hinder the adoption of existing methods. Using Prolog's logic-based programming, the study successfully integrates warehouse, product, and forklift dimensions into a dynamic model, ensuring safe and efficient layout configurations. The proposed solution offers a simple yet effective way to optimise warehouse space while maintaining safety standards for forklift operations. Validation through various warehouse setups demonstrates the model's adaptability and effectiveness across different sizes and configurations. The framework developed in this study bridges the gap between complex mathematical models and the practical needs of warehouse workers, contributing to the optimisation of warehouse layouts, enhancing overall operational efficiency, and supporting sustainable organisational needs. Future research will focus on expanding the model by developing a web-based 2D and 3D simulation platform to provide more interactive and visual tools for warehouse layout design. This simulation model will allow users to experiment with different warehouse configurations, product sizes, and forklift specifications in a user-friendly environment. Additionally, incorporating real-time data analytics and integration with Internet of Things (IoT) devices could further enhance the adaptability and efficiency of the layout optimisation process. Further development of AI-driven optimisation algorithms could automate decision-making processes, making warehouse layout design even more efficient, responsive to operational needs, and aligned with long-term sustainability goals.

## REFERENCES

[1] M. Van Geest, B. Tekinerdogan, and C. Catal, "Design of a reference architecture for developing smart warehouses in industry 4.0," *Comput. Ind.*, vol. 124, p. 103343, 2021.

[2] A. Buzu, "The effect of Warehousing management on Warehouse performance," *Available SSRN 3951785*, 2021.

[3] H. Fatorachian and H. Kazemi, "Impact of Industry 4.0 on supply chain performance," *Prod. Plan. Control*, vol. 32, no. 1, pp. 63–81, 2021.

[4] S. Kumar, B. E. Narkhede, and K. Jain, "Revisiting the warehouse research through an evolutionary lens: a review from 1990 to 2019," *Int. J. Prod. Res.*, vol. 59, no. 11, pp. 3470–3492, 2021.

[5] B. Unhelkar, S. Joshi, M. Sharma, S. Prakash, A. K. Mani, and M. Prasad, "Enhancing supply chain performance using RFID technology and decision support systems in the industry 4.0–A systematic literature review," *Int. J. Inf. Manag. Data Insights*, vol. 2, no. 2, p. 100084, 2022.

[6] E. K. Leung, C. K. H. Lee, and Z. Ouyang, "From traditional warehouses to Physical Internet hubs: A digital twin-based inbound synchronization framework for PI-order management," *Int. J. Prod. Econ.*, vol. 244, p. 108353, 2022.

[7] M. Jiang and G. Q. Huang, "Intralogistics synchronization in robotic forward-reserve warehouses for e-commerce last-mile delivery," *Transp. Res. Part E Logist. Transp. Rev.*, vol. 158, p. 102619, 2022.

[8] N. Jamili, P. L. Van Den Berg, and R. De Koster, "Quantifying the impact of sharing resources in a collaborative warehouse," *Eur. J. Oper. Res.*, vol. 302, no. 2, pp. 518–529, 2022.

[9] H. B. Rai *et al.*, "'Proximity logistics': Characterizing the development of logistics facilities in dense, mixed-use urban areas around the world," *Transp. Res. Part Policy Pract.*, vol. 166, pp. 41–61, 2022.

[10] K. R. Gue and R. D. Meller, "Aisle configurations for unit-load warehouses," *IIE Trans.*, vol. 41, no. 3, pp. 171–182, 2009.

[11] K. J. Roodbergen, I. F. Vis, and G. D. Taylor Jr, “Simultaneous determination of warehouse layout and control policies,” *Int. J. Prod. Res.*, vol. 53, no. 11, pp. 3306–3326, 2015.

[12] E. Ernawati, D. Suprayitno, Y. Evitha, and W. Latunreng, “The Effect of Warehouse Layout on Work Productivity at PT Perkasa Primarindo,” *Int. J. Environ. Sustain. Soc. Sci.*, vol. 4, no. 1, pp. 94–114, 2023.

[13] C.-L. Yang and T. P. Q. Nguyen, “Constrained clustering method for class-based storage location assignment in warehouse,” *Ind. Manag. Data Syst.*, vol. 116, no. 4, pp. 667–689, 2016.

[14] T. Van Gils, K. Ramaekers, A. Caris, and R. B. 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, 2018.

[15] A. S. Dijkstra and K. J. Roodbergen, “Exact route-length formulas and a storage location assignment heuristic for picker-to-parts warehouses,” *Transp. Res. Part E Logist. Transp. Rev.*, vol. 102, pp. 38–59, 2017.

[16] N. V. Kumar and C. S. Kumar, “Development of collision free path planning algorithm for warehouse mobile robot,” *Procedia Comput. Sci.*, vol. 133, pp. 456–463, 2018.

[17] Z. Wanying, W. Xiaoqing, M. Ruofei, Z. Jianchen, and others, “Route optimization of AGVs with an improved fishbone warehouse layout,” in *IOP Conference Series: Earth and Environmental Science*, IOP Publishing, 2021, p. 012047.

[18] Y. Wan and Y. Liu, “Integrating optimized fishbone warehouse layout, storage location assignment and picker routing,” *IAENG Int. J. Comput. Sci.*, vol. 49, no. 3, pp. 957–974, 2022.

[19] L. M. Pohl, R. D. Meller, and K. R. Gue, “Optimizing fishbone aisles for dual-command operations in a warehouse,” *Nav. Res. Logist. NRL*, vol. 56, no. 5, pp. 389–403, 2009.

[20] L. F. Cardona, L. Rivera, and H. J. Martínez, “Analytical study of the fishbone warehouse layout,” *Int. J. Logist. Res. Appl.*, vol. 15, no. 6, pp. 365–388, 2012.

[21] K. A. Clark and R. D. Meller, “Incorporating vertical travel into non-traditional cross aisles for unit-load warehouse designs,” *IIE Trans.*, vol. 45, no. 12, pp. 1322–1331, 2013.

[22] L. Zhou, J. Liu, X. Fan, D. Zhu, P. Wu, and N. Cao, “Design of v-type warehouse layout and picking path model based on internet of things,” *IEEE Access*, vol. 7, pp. 58419–58428, 2019.

[23] L. Zhou, Z. Li, N. Shi, S. Liu, and K. Xiong, “Performance analysis of three intelligent algorithms on route selection of fishbone layout,” *Sustainability*, vol. 11, no. 4, p. 1148, 2019.

[24] X. Zhang, T. Mo, and Y. Zhang, “Optimization of Storage Location Assignment for Non-Traditional Layout Warehouses Based on the Firework Algorithm,” *Sustainability*, vol. 15, no. 13, p. 10242, 2023.

[25] D. Seipel, F. Nogatz, and S. Abreu, “Domain-specific languages in Prolog for declarative expert knowledge in rules and ontologies,” *Comput. Lang. Syst. Struct.*, vol. 51, pp. 102–117, 2018.

[26] P. Körner *et al.*, “Fifty years of Prolog and beyond,” *Theory Pract. Log. Program.*, vol. 22, no. 6, pp. 776–858, 2022.

[27] A. Burinskiene, “Optimising forklift activities in wide-aisle reference warehouse,” *Int. J. Simul. Model.*, vol. 14, no. 4, pp. 621–632, 2015.