Mechanical Engineering – Scientific Journal, Vol. **42,** No. 1, pp. 15–26 (2024)
Number of article: 675
Received: May 8, 2024
Accepted: May 22, 2024
In print: ISSN 1857–5293
On line: ISSN 1857–9191
UDC: 658.286.2:658.78]:[004.421:519.863
https://doi.org/10.55302/MESJ24421015v

Original scientific paper

DETERMINING OPTIMAL ORDER PICKING ROUTE IN WAREHOUSES

Anita Vasileva, Igor Gjurkov

Faculty of Mechanical Engineering, "Ss. Cyril and Methodius" University in Skopje, P.O.Box 464, MK-1001Skopje, Republic of North Macedonia
anita.vasileva@mf.edu.mk

A b s t r a c t: The objective of this research paper is to find the optimal order picking route thus improving the operational efficiency of logistics processes. Computing and selecting the best route are crucial for minimizing order completing time and operational costs. To achieve this, an algorithm is developed by use of several warehouse logistics methods. This algorithm is applied to a single warehouse and the paths are computed using the s-method, the return method, the middle point method, and the composite method. The verification of the algorithm is conducted through thirty cases, each with different order picking locations. This research shows that when larger number of parts need to be picked from various racks, the paths created by the composite method and the s-method provide the shortest route for the designated warehouse.

Key words: path-optimization; order picking; warehouse

ОДРЕДУВАЊЕ ОПТИМАЛНА ПАТЕКА ЗА ПОДГОТОВКА НА НАРАЧКА ВО МАГАЦИНИ

А п с т р а к т: Истражувачкиот труд има за цел да ја пронајде оптималната патека при подготовка на нарачка за испорака, а со тоа да ја подобри ефикасноста на логистичките процеси. Пресметката и изборот на оптималната патека е од суштинско значење за минимизирање на времето за подготовка на нарачка и оперативните трошоци. За да се постигне тоа, развиен е алгоритам користејќи неколку методи на логистика во складиштата. Овој алгоритам е применет на едно складиште и патеката е пресметувана според ,,***s***-методот", ,,методот на враќање", ,,методот на средна точка" и ,,композитниот метод". Верификација на извршниот алгоритам е извршена преку пресметка на триесет случаи кои вклучуваат различни локации во складиштето од кои треба да се земе стока. Истражувањето покажува дека патеките формирани според ,,композитниот метод" и ,,***s***-методот" во сите случаи кога е потребно да се земаат голем број кутии од различни рафтови ја даваат најкратката патека за конкретното складиште.

Клучни зборови: оптимизација на патека; подготовка на нарачка за испорака; складиште

1. INTRODUCTION

The aim of logistics is consistently centered on finding faster, accurate, timely, and flexible methods for delivery. This research paper focuses on the specific aspect of logistics known as intra-logistics. The need for research is supported by a review of earlier surveys in the field of intra-logistics, order picking and optimization of logistics processes in warehouses.

Firstly, Ratliff [1] introduces an algorithm for order picking in warehouses. The objective of the algorithm is to select the optimal path, thereby minimizing the time required for order picking. Kruithof [2] in his research outlines the steps on how to programme an algorithm aimed at optimizing warehouse path. Additionally, Wang [3] in his thesis used a mathematical model, which is based on the Traveling Salesman Problem (TSP), combined with Genetic Algorithms. Puka [4] conducts a comparative analysis of four order picking methods to determine the optimal approach. Liu [5], similarly, presents a comparison of the s-method, the return method, and the composite method. In alignment with previous research, Esra [6] compares the s-shape method, the midpoint method, and the largest gap method.

Lanza [7], in the research paper, gives a thorough explanation of how to optimize the path for order picking using a programming language. Korbacher [8] also seeks to compare paths using the midpoint method, the return method, and the largest gap method. Additionally, Shetty [9] introduces vehicle routing based on order picking in warehouses to further reduce travel time and distance.

The literature review shows that the selection of the shortest path during order picking has been extensively discussed since the second half of the twentieth century. Given that approximately 50% of the time needed for the preparation of a delivery order is attributed to the movement of the order picker, it becomes necessary to optimize the path taken during the execution of this process, thereby reducing the time spent. Within the realm of logistics, order preparation emerges as a critical segment with significant potential for optimizing logistics cost and enhancing productivity [10].

Based on the conducted research, it is evident that addressing the challenge of selecting the shortest path in a warehouse is particularly complex due to the varied shapes and internal layouts of warehouses. Consequently, the primary objective of this study is to identify and capitalize on an opportunity to optimize a pivotal segment of the logistics process-order picking. The focus of this research was to gather data and generate an algorithm for path optimization in the warehouse. Generating and using a Matlab code, this research should efficiently enable the selection of the shortest path in order picking within a certain warehouse. This research paper contributes in solving important and contemporary issues in scientific and professional research into intra-logistics.

2. METHODOLOGY

The data used in this paper was primarily gathered from academic literature and publications. The keywords used in the search are: order picking methods and optimization of pick path in warehouse. Figure 1 illustrates the steps taken in this research.

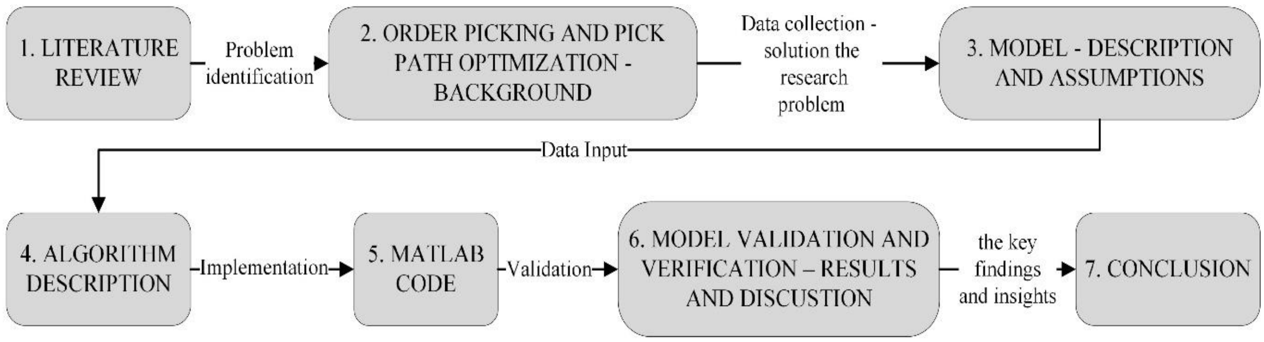

Fig. 1. Research methodology

3. ORDER PICKING METHODS AND PICK PATH OPTIMIZATION

Optimizing order picking involves determining the sequence for order completion and selecting the most suitable path. In the context of rack warehouses, various heuristic methods are employed to generate movement routes. Additionally, algorithms are utilized to ascertain the most appropriate path, whether it be optimal or suboptimal [11].

3.1. *Order picking methods*

a) *S-shape method*

Moving the order picker using the s-shape method is the most straight-forward method to implement. Using this method every rack containing at least one item for retrieval is traversed along its entire length [12, 13]. Conversely, racks devoid of items slated for collection are entirely bypassed. Figure 2 illustrates an example of preparing a delivery order using the s-shape method.

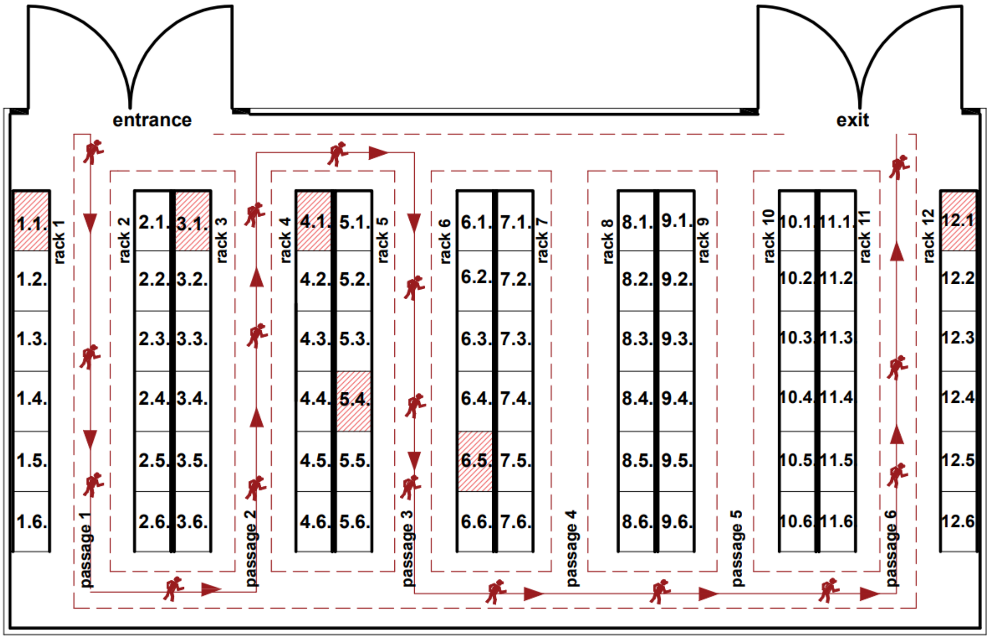

Fig. 2. S-shape method

b) *Return method*

The return method stands out as a simple and frequently employed approach in order picking. In this method, the order picker enters between the racks, exclusively from one side. After collecting the required goods, the order picker follows the same route for the return journey (Figure 3). It's important to note that this method might not always be practical, especially in situations where order picking is done with forklifts and there is not enough room to perform a 180˚ turn [7]. A further drawback of this method is its tendency to prolong the duration of the process compared to alternative methods, resulting in limited optimization of the time allocated for preparing the order for delivery.

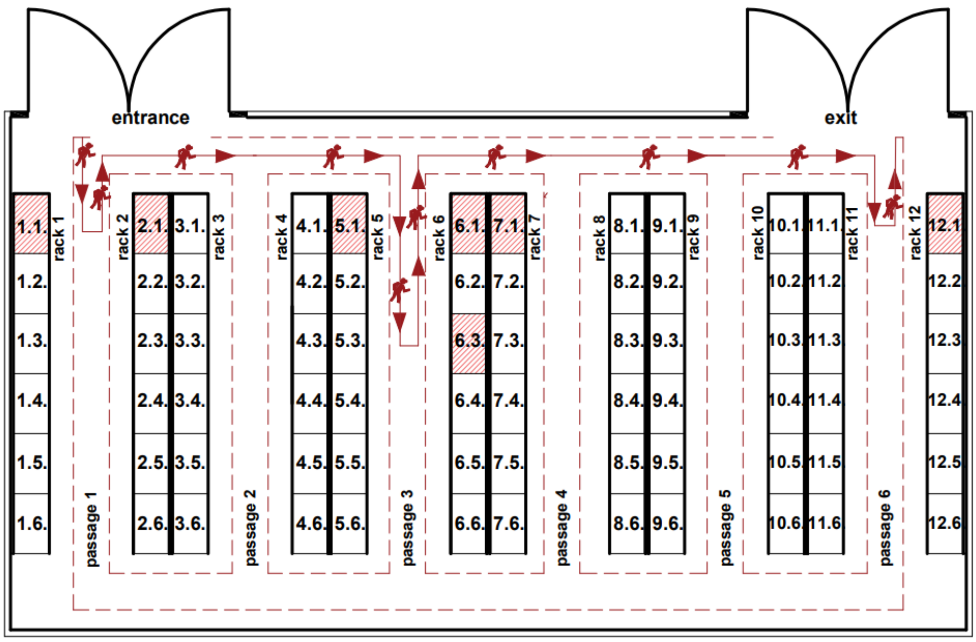

Fig. 3. Return method

c) *Midpoint method*

In this method, the warehouse is conceptually divided into two hypothetical halves. Thus, the completion of the order is done by entering from the first passage and only the first and last passage are traversed across their entire length. When passing through the remaining passages, an alternative approach is adopted. If the items to be picked are situated in either the first or second hypothetical half of the warehouse, the return method is applied. Access to these passages is determined by the corresponding side. For instance, when goods need to be retrieved from the second half of the warehouse (rack bays 4, 5, or 6), entry is made from that side (Figure 4). In contrast, when goods are to be picked from the first half of the warehouse (rack bays 1, 2, or 3), the access point for these passages is from the front side of the warehouse. When passages do not contain items for picking, they are simply bypassed. This approach is illustrated in Figure 4, showcasing how the return method is applied selectively depending on where the parts are located in the two hypothetical warehouse halves [8].

It's crucial to note that the midpoint method might not apply in all situations and could encounter challenges in cases where there's an odd number of rack bays in a sequence. In such cases, decisions must be made on how to divide the racks across the two warehouse halves.

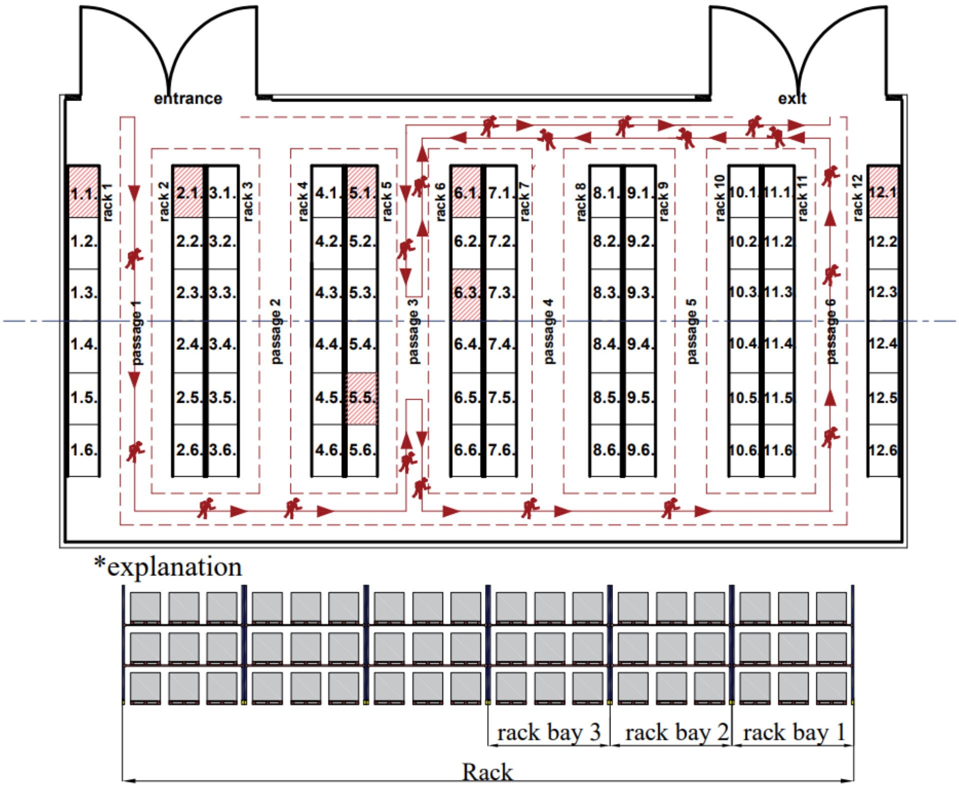

Fig. 4. Midpoint method

d) *Composite method*

As the name suggests, the composite method represents a new method that is a combination of the two existing methods (the s-shape method and the return method), which pulls out the best features of both methods and combines them into one composite version. The purpose of this method is to minimize the movement between two farthest locations and two adjacent crossings, so the aim is to use either the s-shape method or the return method [5, 14]. A composite method example is provided in Figure 5.

Since there are parts to be taken from the bays in the first and the second half (racks bays 3.6 and 4.3) the s-method is applied, for picking the parts in those two passages, the first and the second passage are passed in their entire length. The return method is applied in the third passage, because in that passage the only parts to be taken are the ones that are located in the first half of the warehouse (rack bays 5.1, 6.1 and 6.3). The passages 4 and 5 are bypassed, since there are no parts to be taken.

Unlike the s-method, in the composite method the last passage does not have to be passed in its entire length. Therefore, as shown in Figure 5, the last passage, passage 6, is not passed in its entire length, since it is necessary to take goods from the first bay of the rack (12.1).

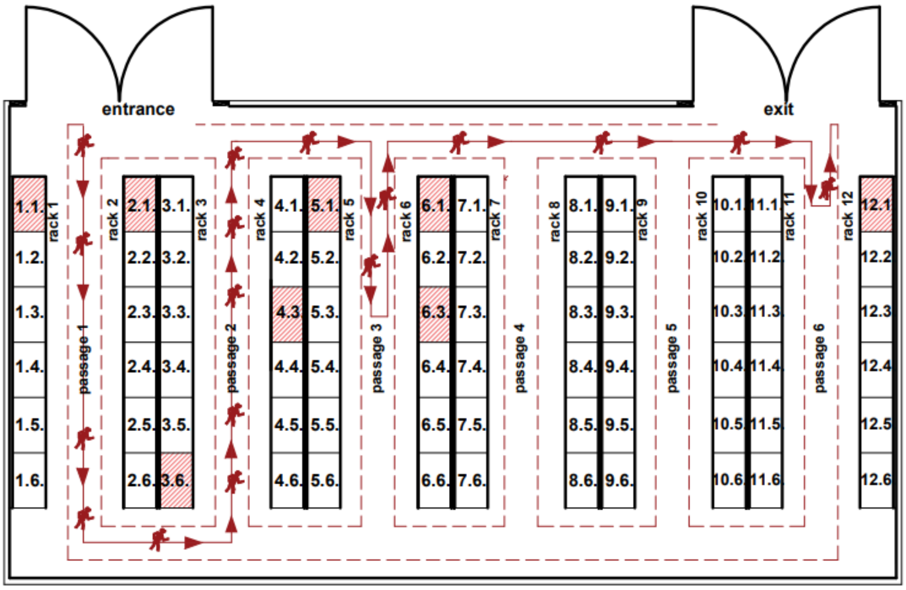

Fig. 5. Composite method

3.2. *Methods of pick path optimization*

The basic approach for calculating the direct distance between two given points, $A(x_1, y_1)$ and $B(x_2, y_2)$ or Euclidean distance, is calculated according to the equation (1):

$$d(A,B) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}. \quad (1)$$

Manhattan distance is another way of distance calculation. According to the Manhattan distance, the distance between two points $A(x_1, y_1)$ и $B(x_2, y_2)$ is calculated using the expression (2) [15]:

$$d(A,B) = |x_1 - x_2| + |y_1 - y_2|. \quad (2)$$

In the literature review on route optimization during the preparation of a delivery order, the Steiner Traveling Salesman Problem (STSP) is used, which is a subtype of the traditional TSP [13, 15]. The TSP deals with scenarios where a known number of locations $\{l_1, l_2,...,l_n\}$ are given, and the traveling salesman must visit each location, considering the distances between each pair of locations [2, 16, 17]. The objective is to find the shortest path along which the traveling salesman will visit all locations and return to the starting location. The spatial solution comprises the set of all permutations of the numbers 1, 2,...,n without repetition. Some of the methods for calculating TSP include the Brute-force method and the Branch and Bound method [18].

If the task involves determining routes for the movement of vehicles, it is known as the Vehicle Routing Problem (VRP) [19, 20]. This method expands upon the TSP. In the TSP, the goal is to visit a specific number of locations from the starting point and ultimately return to the starting point [1]. The fundamental form of the VRP focuses on optimizing the routes for vehicles originating from a central location, delivering goods to various destinations [21]. This could involve the use of several types of vehicles or a single vehicle making multiple trips [9, 22, 23]. The objectives of determining vehicle movement routes include:

- Minimizing the total distance traveled;
- Ensuring that the selected path passes through all locations only once;
- Ensuring that the chosen route starts and ends at the same location.

The objective of the Steiner Traveling Salesman Problem (STSP) [2, 7, 8, 10, 12, 24, 25] is, given a list of locations to be visited and the mutual distances between them, to find the shortest possible path. This path should encompass all locations, ultimately returning to the initial position [21, 26, 27].

4. MODEL DESCRIPTION

This section illustrates the warehouse model used in this research paper. First, the model under study is described in detail. To examine a real-world scenario, input parameters for an actual warehouse are employed. The layout of the warehouse under consideration is depicted in Figure 6. The dimensions of the warehouse are 31,900 mm in width and 11,000 mm in length. The warehouse consists of twelve racks, five of which being double-sided racks and two of which being single-sided racks. Each rack has six bays. The length of each rack is 6,000 mm, while each rack bay is 1,000 mm in length. The distance between the racks and the warehouse walls is 2,500 mm, and the gap between two adjacent racks is 2,800 mm. The width of the single-sided racks is 1,300 mm, and for the double-sided racks is 2,500 mm. The warehouse features a single entrance and exit, both located on the same side of the building. The entrance is positioned on the leftmost side, while the exit is on the right. Within the warehouse, six passages between the racks allow for the movement and manipulation of goods using a forklift. In Figure 6, the path anticipated during order creation is marked in red. In detail A of Figure 6, the 1,000 mm distance represents the space between two rack bays of one rack when collecting parts. In detail B of Figure 6, the 500 mm distance indicates the space from the beginning of the rack to the middle of the first rack bay.

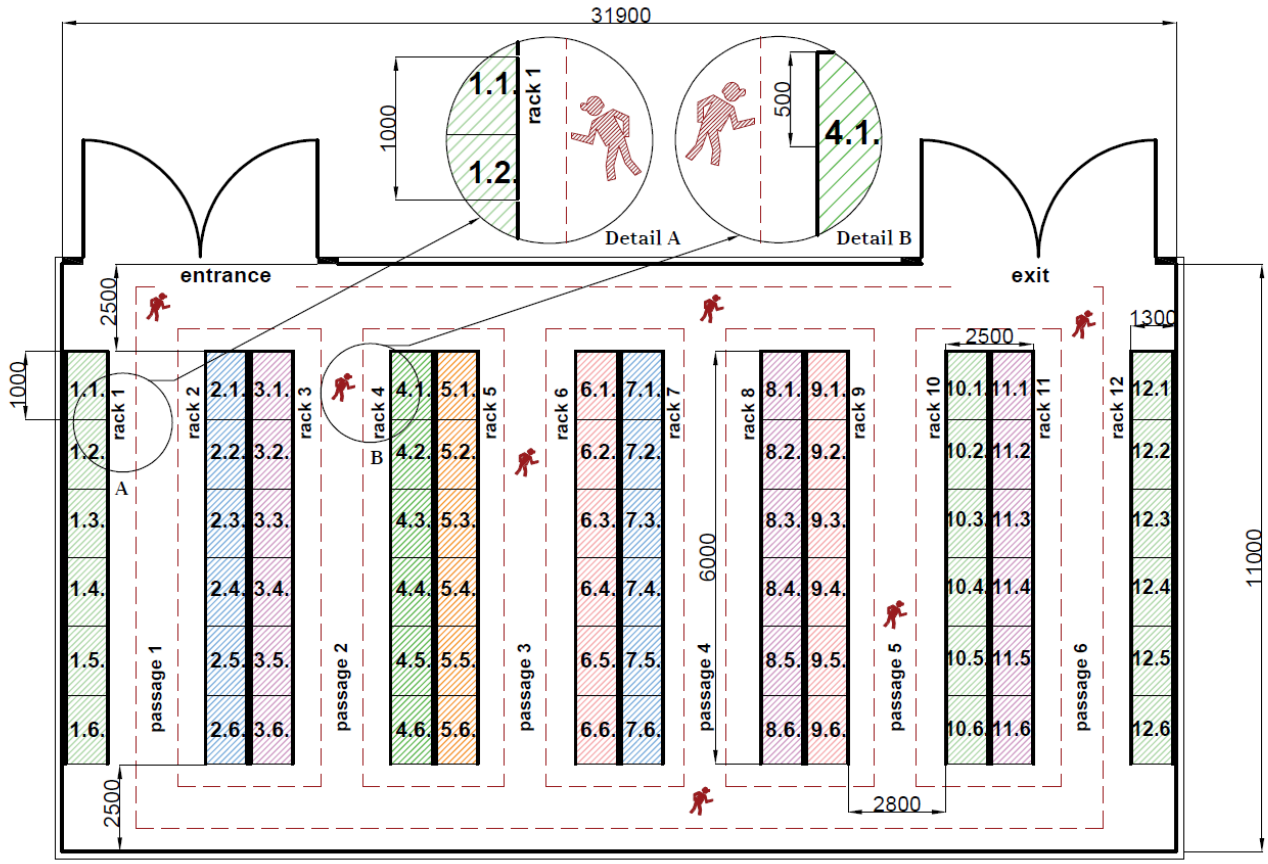

Fig. 6. Warehouse dimensions

For a simpler representation of the warehouse in Figure 6, the racks and rack bays are labeled. The racks are numbered from 1 to 12, and each rack bay is labeled from 1 to 6. Therefore, for rack 1, the rack bays are marked as 1.1, 1.2, 1.3, where the first number indicates the rack number, and the second number indicates the bay number on that rack.

The height of both the racks and the warehouse are not considered, since they are not taken into account in the calculation of the route for order picking.

Additionally, it is expected that the order picker will select boxes from the center of each rack bay to form the delivery order. This research paper omits the dimensions and mass of the boxes and there are no limits on capacity, i.e., on how many parts the order picker can carry.

4.1. *Presentation of the algorithm*

Figure 7 illustrates the block diagram of the algorithm for creating an executable program. The algorithm operates by calculating paths using four methods and choosing the best one. The first step of the algorithm is to create a matrix with 12 columns and 6 rows, where the number of columns corresponds to the number of racks, and the number of rows represents the bays of each rack.

In the next step, the code generates a matrix with all elements initialized to zero. To modify the matrix, the positions from which an order needs to be picked are changed from zero to one. This is achieved using the command 'change = 1:num_changes', where the variable ‘num_changes’ denotes the number of orders to be taken. The cycle iterates 'n' times, prompting the user to specify from which rack (column) and rack’s bay (row) packet need to be taken. After completing the ‘for’ operation, the matrix is formed, with ‘1’ representing the positions from which orders should be picked. Following this, the output matrix is displayed, highlighting the locations marked with '1' as the places from which goods should be retrieved.

After the completion of the previous steps, the algorithm proceeds to calculate the sub-processes for the entered parameters based on the four methods: the s-method, the return method, the midpoint method, and finally, the composite method. Following the execution of these four sub-processes, the algorithm compares the path lengths obtained from each method. If the length of the path according to the s-method is smaller than the lengths obtained from the other three methods, the output is printed: "The path according to the s-method is the shortest". However, if the path length according to the s-method is not smaller than at least one of the paths obtained from the other three methods, a comparison is made with the length of the path obtained according to the return method, the midpoint method, and the composite method. If the length obtained from the return method is the smallest compared to the other three methods, the output is: "The path according to the return method is the shortest". In case the length obtained from the midpoint method is the smallest compared to the other three methods, the output is: "The path according to the mid-point method is the shortest". Otherwise, the output is: "The path according to the composite method is the shortest".

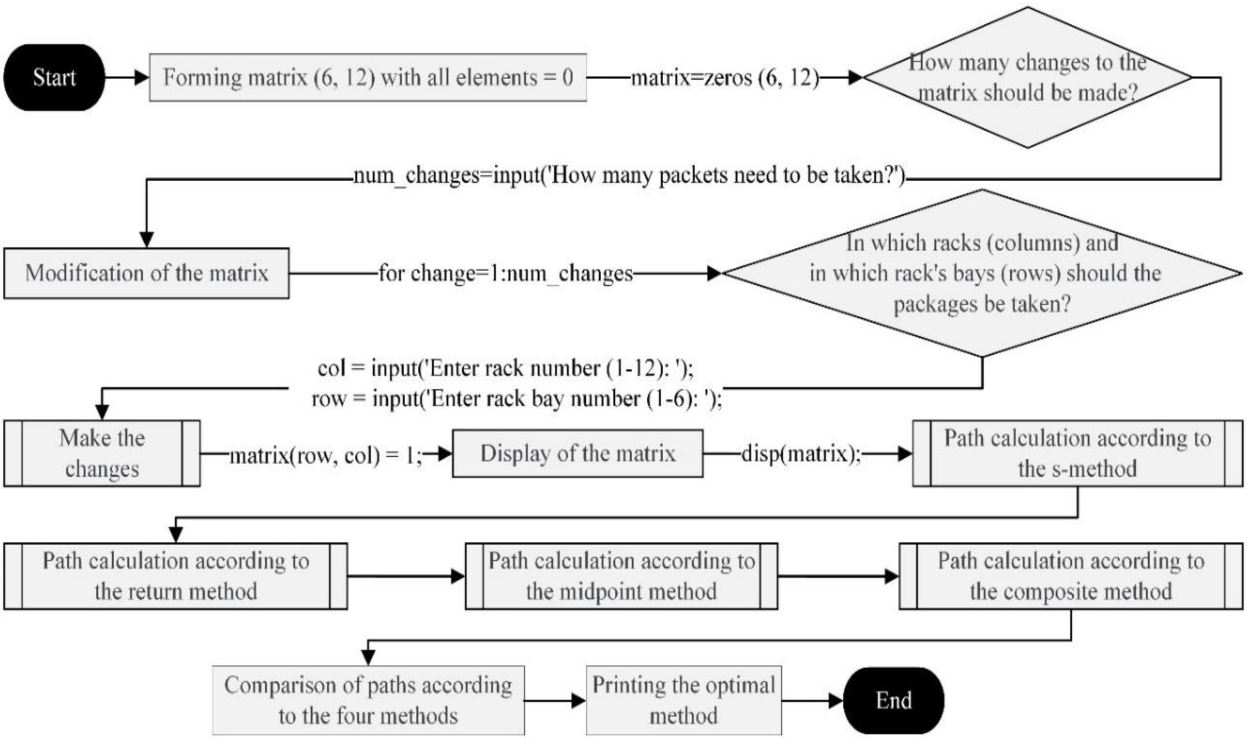

Fig. 7. Block diagram of the algorithm

5. MODEL VALIDATION AND VERIFICATION

Results and discussion

Thirty distinct arbitrary order picking cases were generated to evaluate the reliability of the Matlab code. To validate the code, three cases were randomly selected from the total of thirty and manually calculated. In Figure 8, the slice window for case 1 is displayed, while Figure 9 shows the paths generated according to the four methods for the input data of case 1.

In Figure 10, the slice window for case 2 is shown, while Figure 11 presents the paths generated according to the four methods for the input data of case 2.

In Figure 12, the slice window for case 3 is shown, while Figure 13 presents the paths generated according to the four methods for the input data of case 3.

```
Command Window
View of the warehouse with the packages that need to be taken when forming the order:
     1     1     1     1     1     1     1     1     1     1     0     0
     1     0     0     1     1     0     1     0     0     0     0     1
     1     1     0     0     1     0     0     1     1     1     0     0
     1     1     0     1     1     0     1     0     0     1     1     1
     0     1     1     0     0     0     1     1     0     0     0     1
     0     0     1     1     0     1     1     1     1     0     1     0


The path length by the S-method is: 80000
The path length by the Return method is: 108000
The path length by the Midpoint method is: 143900
The path length by the Composite method is: 80000
The path is the shortest according to this/those method/s: The S-method 80000, The composite method 80000
fx >>
```

Fig. 8. Case 1

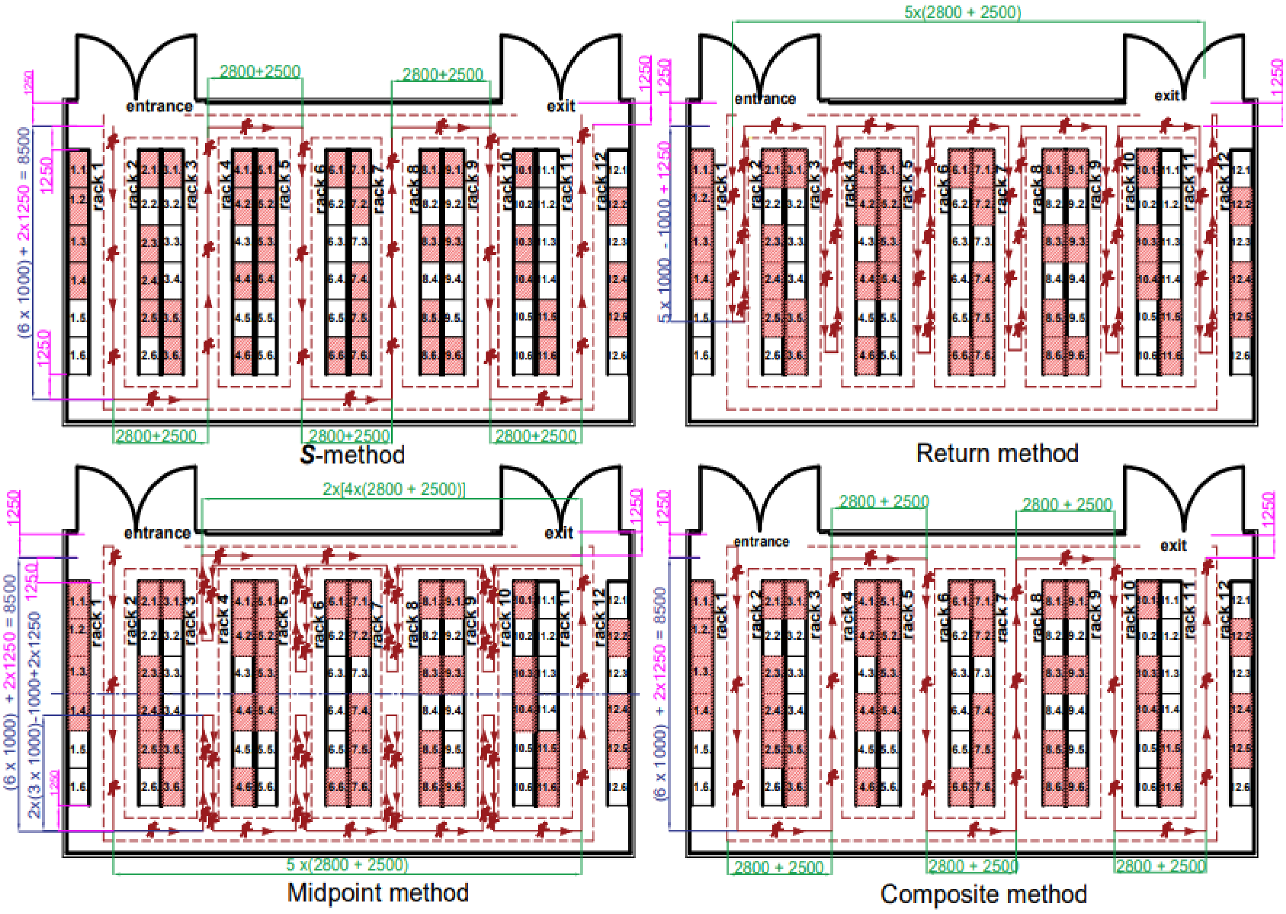

Fig. 9. Comparison of path lengths according to the four methods (Case 1)

```
Command Window
View of the warehouse with the packages that need to be taken when forming the order:
     0     1     1     1     0     1     1     1     1     1     1     0
     0     1     0     0     0     1     0     1     1     0     0     1
     0     0     1     0     1     0     1     1     1     1     1     0
     0     1     1     0     1     0     1     1     0     1     0     0
     0     0     1     1     1     0     0     0     1     0     1     0
     1     1     1     1     0     1     0     0     1     1     0     0

The path length by the S-method is: 80000
The path length by the Return method is: 104000
The path length by the Midpoint method is: 141900
The path length by the Composite method is: 80000
The path is the shortest according to this/those method/s: The S-method 80000, The composite method 80000
fx >>
```

Fig. 10. Case 2

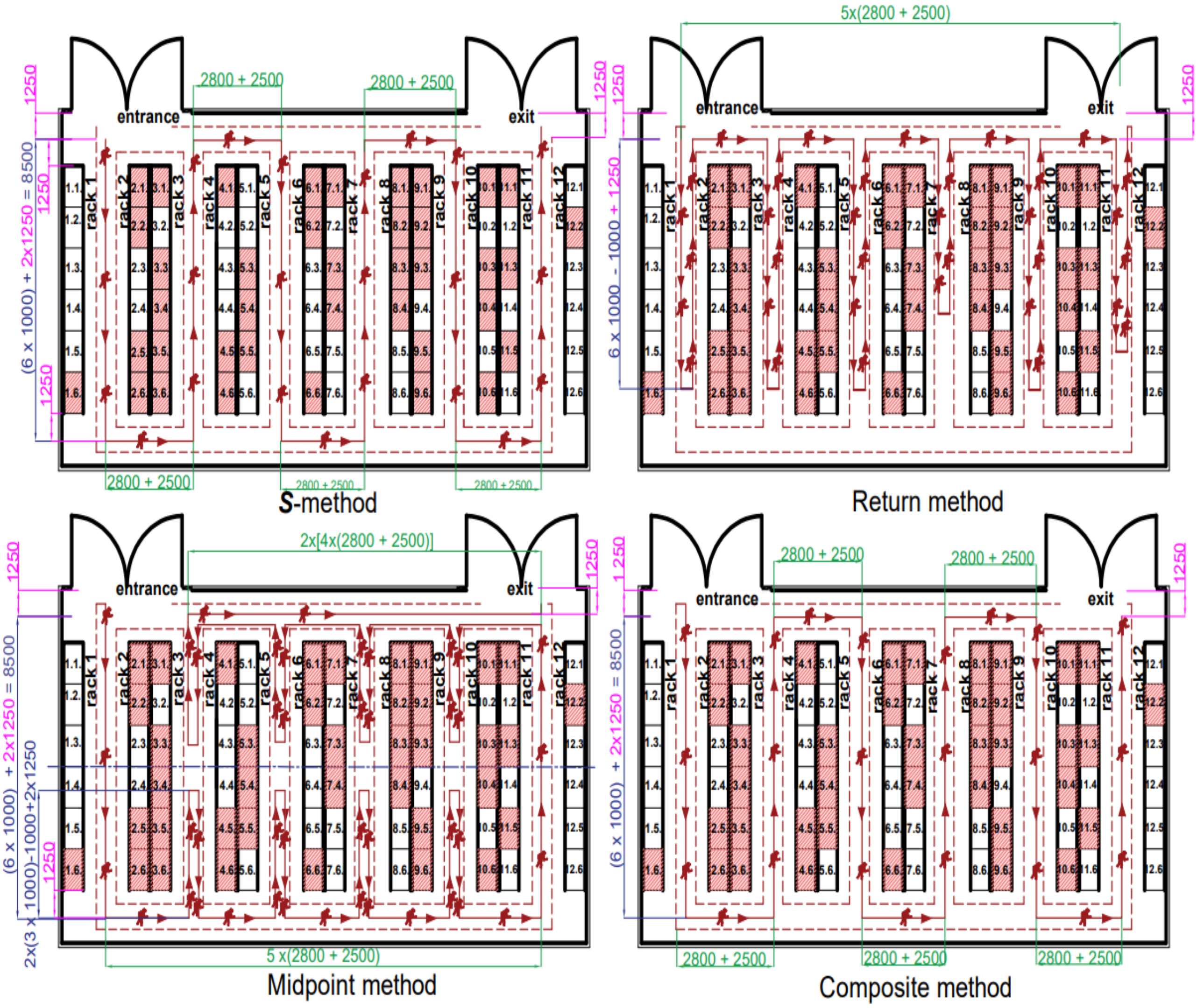

Fig. 11. Comparison of path lengths according to the four methods (Case 2)

```
View of the warehouse with the packages that need to be taken when forming the order:
      0     0     1     0     0     0     1     0     0     0     0     1
      1     0     0     0     0     0     0     0     0     0     0     0
      0     0     0     0     0     0     0     0     0     0     0     0
      0     0     0     0     0     0     0     0     0     0     0     0
      0     0     0     0     0     0     0     0     0     0     0     0
      0     0     0     0     0     0     0     0     0     0     0     0

The path length by the S-method is: 53000
The path length by the Midpoint is: 44000
The path lengh by the Midpoint method is: 90400
The path length by the Composite method is: 44000
The path is the shortest according to this/those method/s: The return method 44000, The composite method 44000
```

Fig. 12. Case 3

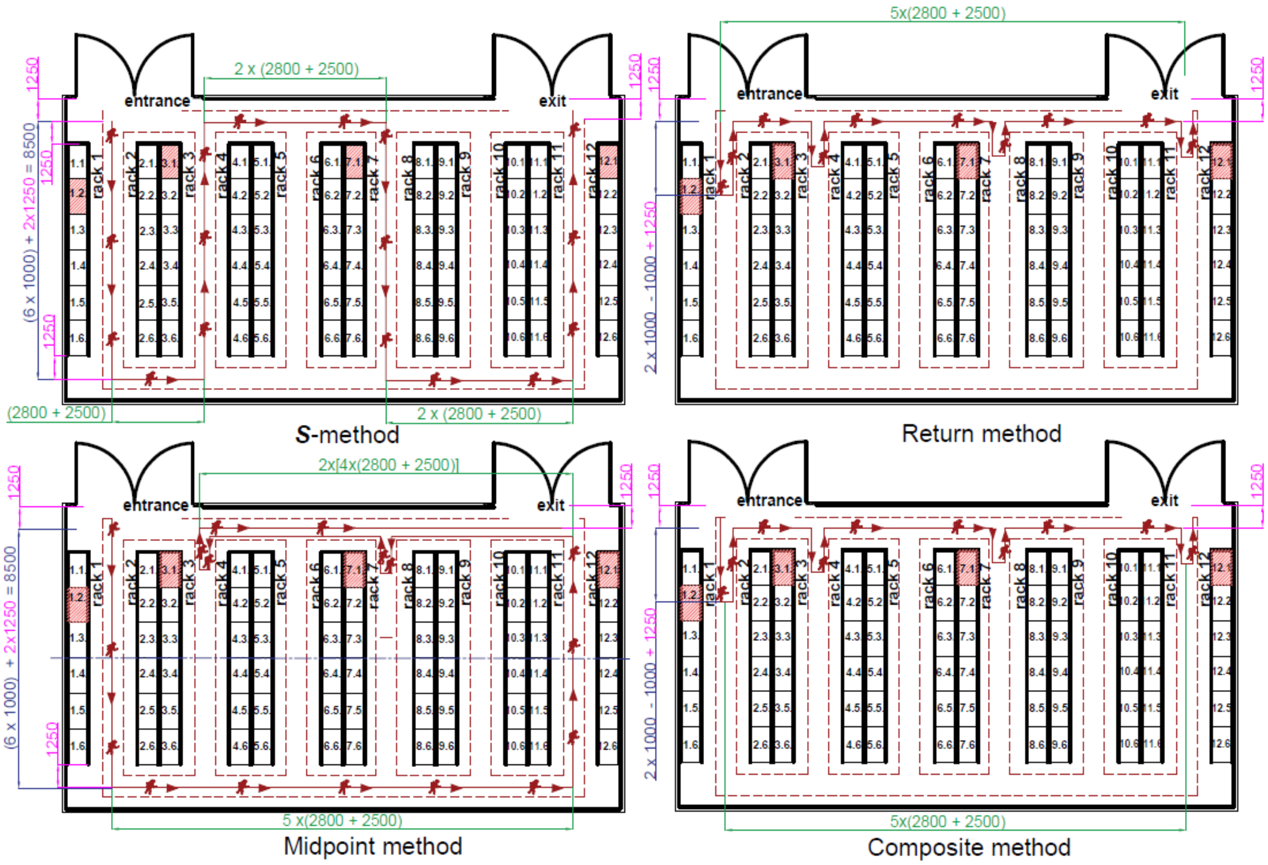

Fig. 13. Comparison of path lengths according to the four methods (Case 3)

Based on the results obtained, the formation of paths for order picking using the composite method and the s-method consistently outperformed in all of the test cases compared to the other methods. The identical values of the paths for the composite method and the s-method arise from the nature of the orders in the illustrated cases. These cases require retrieving a significant number of parts/boxes from nearly every bay of the racks, resulting in the formation of the same path for both the composite method and the s-method. In all cases, it is worth noting that the lengths according to the s-method are consistently 80,000 mm. This uniformity in path lengths for different orders stems from the necessity to retrieve a box from each rack, requiring passage through all the passages to enable access to each rack bay. After the composite method and the s-method, the next best-performing method is the return method. The longest paths in all thirty cases, are observed with the midpoint method. For the given warehouse and its entry and exit layout, the midpoint method fails to provide an adequate path formation in order picking. In scenarios where only a small number of boxes need to be retrieved from the first bays of the racks, the return method, as well as composite method, yields the shortest path.

6. CONCLUSIONS

The selection of the shortest delivery path is a logistical challenge that has been extensively studied since the second half of the 20th century. In logistics, the preference for the shortest path is primarily applied during the delivery of shipments within the borders of a country, for intercity or city deliveries. The issue of choosing the shortest path when preparing goods for delivery within the warehouse, as part of the logistics chain, is a problem that is not as thoroughly researched, and few executable programs are available. The varying shapes and internal layouts of warehouses add to the difficulty of solving the issue of choosing the shortest path within a warehouse. This suggests that a software solution needs to be tailored to the placement of input and output, as well as the layout of the racks where the items are stored. Selecting the optimal path when preparing an order for delivery in a warehouse is essential for optimizing logistics processes. In this regard, this research paper addresses the specific problem of calculating the shortest path in a rack warehouse with given dimensions and layout.

The verification of the executable code is demonstrated through thirty arbitrary order-picking cases. The simulation results show that for varying numbers of racks and rack's bays from which part/boxes need to be picked, the composite method and the s-method show superior performance among the four methods. As expected, the composite method outperforms the other techniques and provides optimal solutions by combining and addressing the limitations of alternative methods. In scenarios where numerous boxes are to be picked up from almost all racks, the paths formed by the composite method and the s-method are identical.

When it comes to retrieving parts/boxes from a smaller number of racks, particularly from the first or the second rack bays in the first half of the warehouse, the return method yields better results than the s-method. As the number of racks increases, and there is a need to collect boxes from more distant rack bays like 4, 5, and 6, the s-method proves more effective. In the given case, the longest paths during order formation in all examples were observed using the midpoint method. In conclusion, the values obtained from the conducted research underscore the importance of calculating the shortest path in warehouses when preparing a delivery order.

Future directions for refining, expanding, and advancing the research would involve a comparative analysis examining how changes in input and output parameters affect path lengths. Additionally, the objective is to develop an executable algorithm capable of mapping the shortest path as an output, enabling the code to find more practical applications. Developing an algorithm to calculate the shortest path for warehouses with varying dimensions and spatial layouts is a substantial challenge.

A major improvement to the algorithm in the future would involve incorporating capacity considerations into the calculation of the optimal path during the process of preparing a delivery order. This entails including the quantity of items retrieved from a particular rack and considering the limited capacity of the asset, such as a cart or a box, into which the order picker will place the ordered pieces.

REFERENCES

[1] Ratliff, H. D., Rosenthal, A. S. (1983): Order-picking in a rectangular warehouse: a solvable case of the traveling salesman problem. *Operations Research* **31** (3), 507–521.

[2] Kruithof, M. W. (2012): *Traveling Salesman Problem. Comparisons between heuristics, linear and semidefinite programming approximations,* Master's thesis in applied mathematics, University of Groningen, Faculty of Mathematics and Sciences.

[3] Wang, P. G., Qi, X. M., Zong, X. P., Zhu, L. L. (2013): Picking route optimization of automated warehouse based on improved genetic algorithms. *Applied Mechanics and Materials* **411**, 2694–2697.

[4] Puka, R., Duda, J., Karkula, M. (2019, June): Improving of selected routing strategies for order picking. In: *International Scientific Conference Transport of the 21st Century* (pp. 363–373). Cham: Springer International Publishing.

[5] Liu, H., Wang, F., Zhao, J., Yang, J., Tan, C., Zhou, L. (2022): Performance analysis of picking path strategies in chevron layout warehouse. *Mathematics* **10** (3), 395.

[6] Esra, B. O. Z., Nil, A. R. A. S. (2022): Order picker routing problem in a single block warehouse. *Journal of Turkish Operations Management* **6** (1), 1113–1120.

[7] Lanza, G., Passacantando, M., Scutellà, M. G. (2022): Sequencing and routing in a large warehouse with high degree of product rotation. *Flexible Services and Manufacturing Journal*, 1–50.

[8] Korbacher, L., Heßler, K., Irnich, S. (2023): *The single picker routing problem with scattered storage: Modeling and evaluation of routing and storage policies,* Discussion papers No. 2302, Johannes Gutenberg Universität, Mainz.

[9] Shetty, N., Sah, B., Chung, S. H. (2020): Route optimization for warehouse order picking operations via vehicle routing and simulation. *SN Applied Sciences*, **2**, article number 311, pp. 1–18.

[10] Burinskiene, A. (2010): Order picking process at warehouses. *International Journal of Logistics Systems and Management* **6** (2), 162–178.

[11] Cano, J. A., Correa-Espinal, A. A., Gómez-Montoya, R. A. (2017): An evaluation of picking routing policies to improve warehouse efficiency. *International Journal of Industrial Engineering and Management* **8** (4), 229.

[12] Goeke, D., Schneider, M. (2021): Modeling single-picker routing problems in classical and modern warehouses. meritorious paper awardee. *INFORMS Journal on Computting* **33** (2), 436–451.

[13] Zhou, L., Zhao, J., Liu, H., Wang, F., Yang, J., Wang, S. (2022): Stochastic models of routing strategies under the class-based storage policy in fishbone layout warehouses. *Scientific Reports* **12** (1), 12876.

[14] Merkuryev, Y., Burinskiene, A., Merkuryeva, G. (2009): Warehouse order picking process. *Simulation-Based Case Studies in Logistics: Education and Applied Research*, 147–165, Technical University, Riga (Latvia).

[15] Birge, J. R., Louveaux, F. (2010): *Introduction to Stochastic Programming* (Second Edition), Springer. DOI 10.1007/978-1-4614-0237-4.

[16] Lu, W., McFarlane, D., Giannikas, V., Zhang, Q. (2016): An algorithm for dynamic order-picking in warehouse operations. *European Journal of Operational Research* **248** (1), 107–122.

[17] Theys, C., Bräysy, O., Dullaert, W., Raa, B. (2010): Using a TSP heuristic for routing order pickers in warehouses. *European Journal of Operational Research* **200** (3), 755–763.

[18] Brucato, C. (2013): *The Traveling Salesman Problem*. Master's thesis, University of Pittsburgh (Unpublished).

[19] Burkard, R. E., Fruhwirth, B., Rote, G. (1995): Vehicle routing in an automated warehouse: Analysis and optimization. *Annals of Operations Research* **57**, 29–44.

[20] Yang, J., Zhou, L., Liu, H. (2021): Hybrid genetic algorithm-based optimisation of the batch order picking in a dense mobile rack warehouse. *PLoS ONE* **16** (4), e0249543, doi: 10.1371/journal.pone.0249543

[21] Namany, S., Kissani, I. (2017): *Capacitated Vehicle Routing Problem with Variable Fleet of Delivery Vehicles of Uniform Capacity*. Capstone report, University of Al Akhawayn, Morocco.

[22] Roodbergen, K. J., Koster, R. (2001): Routing methods for warehouses with multiple cross aisles. *International Journal of Production Research* **39** (9), 1865–1883.

[23] Shen, X., Yi, H., Wang, J. (2021, March): Optimization of picking in the warehouse. In: *Journal of Physics: Conference Series* Vol. **1861**, No. 1, p. 012100. IOP Publishing.

[24] Bottani, E., Montanari, R., Rinaldi, M., Vignali, G. (2015): *Intelligent algorithms for warehouse management*. Intelligent techniques in engineering management: Theory and applications. 645–667, Springer.

[25] Ivanov, D., Tsipoulanidis, A., Schönberger (2019): *Global Supply Chain and Operations Management*, Second Edition. Springer.

[26] Pinkam, N., Bonnet, F., Chong, N. Y. (2016, October): Robot collaboration in warehouse. In *2016 16th International Conference on Control, Automation and Systems (ICCAS)* (pp. 269–272). IEEE.

[27] Vivaldini, K. C. T., Becker, M., Caurin, G. A. (2009, November): Automatic routing of forklift robots in warehouse applications. In: *Proceedings of the 20th International Congress of Mechanical Engineering*, Gramado-RS, Brazil.