applied sciences

MDPI

Article

PickupSimulo–Prototype of Intelligent Software to Support Warehouse Managers Decisions for Product Allocation Problem

Augustyn Lorenc [1,*] and Tone Lerher [2]

[1] Faculty of Mechanical Engineering, Cracow University of Technology, 31-864 Kraków, Poland
[2] Faculty of Logistics, University of Maribor, Mariborska cesta 7, 3000 Celje, Slovenia; tone.lerher@um.si
* Correspondence: alorenc@pk.edu.pl; Tel.: +48-12-374-36-59

Received: 12 November 2020; Accepted: 3 December 2020; Published: 4 December 2020

Featured Application: The results of the proposed research could be a useful tool in the selection of products classification while planning products allocation in a warehouse. Proposed model, as a developed software, can support the decision and decrease the risk of the inappropriate choice of product location planning. Based on the proposed research and the application of the proposed model Pickup Simulo, we can expect the reduction in the total warehouse costs to be between 10 and 16 per cent. The research results are, therefore, of significant importance for warehouse management.

Abstract: In this paper, a new model supporting decisions about product allocation in an order-picking shelf warehouse is presented. Industry 4.0 pays attention to inciting the processes, self-analysis and self-optimization of the short response time to market changes, and the maximum use of related data. Methods for solving the product allocation problem (PAP) are not enough to meet the requirements of Industry 4.0. The authors present a new approach for solving PAP. The novelty introduced in the model is based on correlated data—products parameters, clients' orders and warehouse layout. The proposed model contains elements of intelligence. The model, after product classification and allocation, analyzes its effectiveness by a simulation of the order-picking process. The application of artificial neural networks (ANN) as a part of the computing model enables the analysis of large data sets in a short time. The presented study has proved the proposed model, both for practical and scientific purposes. Relying on the research results, the total warehouse cost could be reduced by 10 to 16 per cent. With the use of the proposed model, it is possible to predict the effect of future actions before their execution. The model can be implemented in most conventional warehouses to raise the throughput performance of the order-picking process.

Keywords: warehousing; order-picking; product allocation problem; artificial neural networks; product classification methods; performance analysis

1. Introduction

Modern e-commerce allows companies to compete not only on a local but also on a global scale [1]. From the perspective of the global market, it is not important where the product comes from, but how much time and resources it takes for a shipment delivery to reach a target customer. The costs of material handling and final transportation are often a significant part of the total cost of a product, especially for products of relatively small value [2].

In the 21st century, a new ideology, Industry 4.0, was formed. With the growth of technology, the large base of data and customer expectations forced the companies to be more effective.

Modern companies must pay attention to inciting the processes, self-analysis and self-optimization of the short response time to market changes and the maximum use of related data. It is a way for a company to be competitive on the market.

Industry 4.0 is also important for intralogistics in a warehouse. The main process in the warehouse which takes the most time is the products' picking. The effectiveness of this process can be optimized by using new technology, such as robots, voice picking, and augmented reality supported by the better insight of the data from WMS (Warehouse Management System) to reduce the distance of picking route. This distance can be reduced by solving the product allocation problem (PAP).

Proper warehouse system planning and PAP in the warehouse are major challenges for industrial companies. The correct assignment of products in warehouses can reduce the cost of energy, the number of forklifts and order-pickers, along with the lead time. For these reasons, it is extremely important to define the optimal storage locations so as to search for better conditions of profitability and service levels in companies.

In intralogistics, a warehouse is often a bottleneck, which raises the total costs of a product. Consequently, if product allocation planning is performed randomly, it may prove that higher-demand products will be located most distant from the picking and packing zone. In such cases, there is a risk of unnecessary travel time for the order-picking process. Therefore, the loss of time for the order-picking and the increase in order-pickers and material handling systems may occur.

The currents methods for solving PAP are not enough to meet the requirements of Industry 4.0. Currently, these methods enable the grouping of products to different ranks. After classification, they are distributed in the warehouse area. This distribution aims to ensure the shortest time to access products of the highest frequency from the gates/dispatch zone. The typical methods of products classification are methods called ABC, XYZ, EIQ, AHP, and COI Index analyses. Although these methods are still widely used in practice, they do not satisfy the requirements placed on the warehouses nowadays. The above methods are most often performed for one single criterion. However, the mentioned methods can be used to allocate products by different criteria at the same time, but do not allow the analysis of correlations between classified products [3–8].

Regardless of the industry branch, the modernization of the logistic processes carried out within the warehouse is one of the key components of building a competitive advantage. This is especially important for e-commerce companies, being the major reason for the optimization of intralogistics processes in warehouses where many products are picked per hour.

This research gap motivates the authors to propose a new approach based on the collected data in WMS. The proposed model is based on correlated data, such as products parameters, clients' orders and warehouse layout. The proposed model in our paper enables us to select the best product classification methods automatically, so as to find the correlation between the analyzed criteria and results. The best methods (ABC, XYZ, EIQ, AHP) are mixed to achieve even better results. The mixing of the methods is done by an artificial neural network. ANN was used as a part of the computing model for the quick analysis of large amounts of data. The proposed model contains elements of intelligence. The model, after product classification and allocation, analyzes its effectiveness by a simulation of the order-picking process [9–15].

Contribution to the Body of Knowledge

Based on the results of the proposed model, warehouse managers can plan the allocation of products in the warehouse. This allocation will be optimal for such factors as actual demand, the allocation of other products, as well as client orders and products parameters. Managers will reduce the time of planning the product allocation and reduce the distances and order-picking time for products. The usefulness of the proposed model will allow the reducing of the total working time of order pickers, which means that the efficiency of the order-picking process will get higher. Following the simulation results of the real case study, it can be observed that the order-picking time is reduced from 10 to 16 per cent. The real cost-savings of the order-picking time depends on the company.

The proposed model can be implemented as a module in a WMS system, or can work separately. The warehouse managers can use this model to optimize the allocation of a product in the warehouse and reduce order-picking distances, and consequently order-picking times. The proposed version of the model is limited to the homogeneous type of racks or for storing the homogeneous type of pallets on the ground. In future work, it is planned to improve the analytical description of the warehouse structure, with the objective that the model could be used for any type of warehousing. The proposed model uses just one type of transport means, with one velocity profile and with the normal distribution. The newest research papers pay attention to the reorganization of the warehouse with regard to the Internet of Things (IoT) ideology. The researchers [16] analyze the processes in traditional warehouses and modern ones. They propose a business process model of a smart restocking process. The authors pay attention to the optimization of the intralogistics.

The model proposed in this paper uses the clients' orders, the products' demand and the products' properties to optimize the allocation of the product. This model can be implemented as part of the warehouse management system (WMS), or can be used as a separate software. This research is consistent with the ideology of Industry 4.0.

2. Literature Review

The product allocation problem has been analyzed in many research papers in the research community. The interesting solution was proposed by Ahmed et al. [17]. The authors present a new approach to the evaluation of the supplier of products and the allocating of the optimal order quantity from each supplier concerning the ecology and resilience of the transport chain. The obtained weights from the analytical hierarchy process (AHP) and the technique for order of preference by similarity to ideal solution (TOPSIS) were integrated. This integration enables the development of a multi-objective model used as an order allocation planner. The "ε-constraint" model was used to solve the multi-objective optimization problem [18]. The "ε-constraint" algorithm aims to optimize one objective function while considering all other objective functions as constraints. For solving similar problems, the CPLEX software can be used. Ballesteros-Riveros et al. [19] focus on the research on the problem of allocation to the proper area by the criterion of time horizon in a food company. The algorithm identifies the operation area and the corresponding spaces that are required for the allocation of products [19]. Xiaocheng Li et al. [20] propose a novel shipping mechanism "Predictive Shipping", which utilizes demand prediction to arrange a ship before orders are placed. They have formulated and solved a large-scale products allocation problem across warehouses [20]. Zapata-Cortes Julian Andres et al. [21] present a mathematical model for solving the product allocation problem based on the genetic algorithm (GA) [21]. The authors calculate the distances for product order-picking to minimize the routing distances. In 2013, a research team from Italy [22] proposed a mathematical model for the multi-level product allocation problem with compatibility constraints. The goal of this research was to reduce, as much as possible, the delivery times, the inventories and the total logistic costs, and to guarantee, at the same time, higher service levels. Researchers used the mathematical modeling sensitivity analysis, then iterated local search-based heuristic was used to reduce the computing time. Two years later, a similar team compared the heuristics for solving the product allocation problem in the multi-level warehouse. This authors present a rollout-based heuristic whose performances are evaluated on the basis of a detailed computational phase, including also a real case study and comparing it with iterated local search-based heuristic and cluster-based heuristic [23].

Many research studies focus on the allocation of a product in the supply chain network. For example, Soleimaninanadegany Amin et al. [24] use a GA for distribution centers, considering processing cost, demand, capacity and related constraints [24]. In 2003, Gengui et al. [25] proposed using the genetic algorithm to allocate a set of customers to multiple warehouses. The proposed solution procedure utilizes a genetic algorithm that is designed to find Pareto optimal solutions for this problem in a short period. Shan Zhu et al. [26] proposed the method for the optimization of product category allocation in multiple warehouses to minimize the splitting of online supermarket customer orders. Multi-item customer orders often need to be split into multiple shipments because the ordered items may be stored in different warehouses. The method developed by the authors used a heuristic clustering algorithm to optimize the product category allocation among multiple warehouses based on the distribution of multi-item orders to minimize the total number of order splits.

According to a literature review, there is a lack of models to support the product classification and allocation problem in modern warehouses. The available literature and market solutions allow for simulating the order-picking process and efficiency analysis. Neither of them, however, allow the easy planning of product allocation in warehouses. Therefore, a new model for product allocation in a shelf order-picking warehouse has been proposed in our research study.

This paper is organized as follows. The paper begins in Section 1 with an introduction to the problem domain, while the literature review on product classification is presented in Section 2. Section 3 concerns a model for the classification of products and the efficiency analysis of order-picking. Section 4 presents the implementation of the described model in the framework of the simulation method. In Section 5 the performance of the products classification and allocation model is evaluated based on the case study. Section 6 includes the main conclusions and discussions.

3. Presentation of the Proposed Model

3.1. General Assumptions

The following assumptions regarding the type of warehouse have been considered for the development of the proposed model for product allocation:

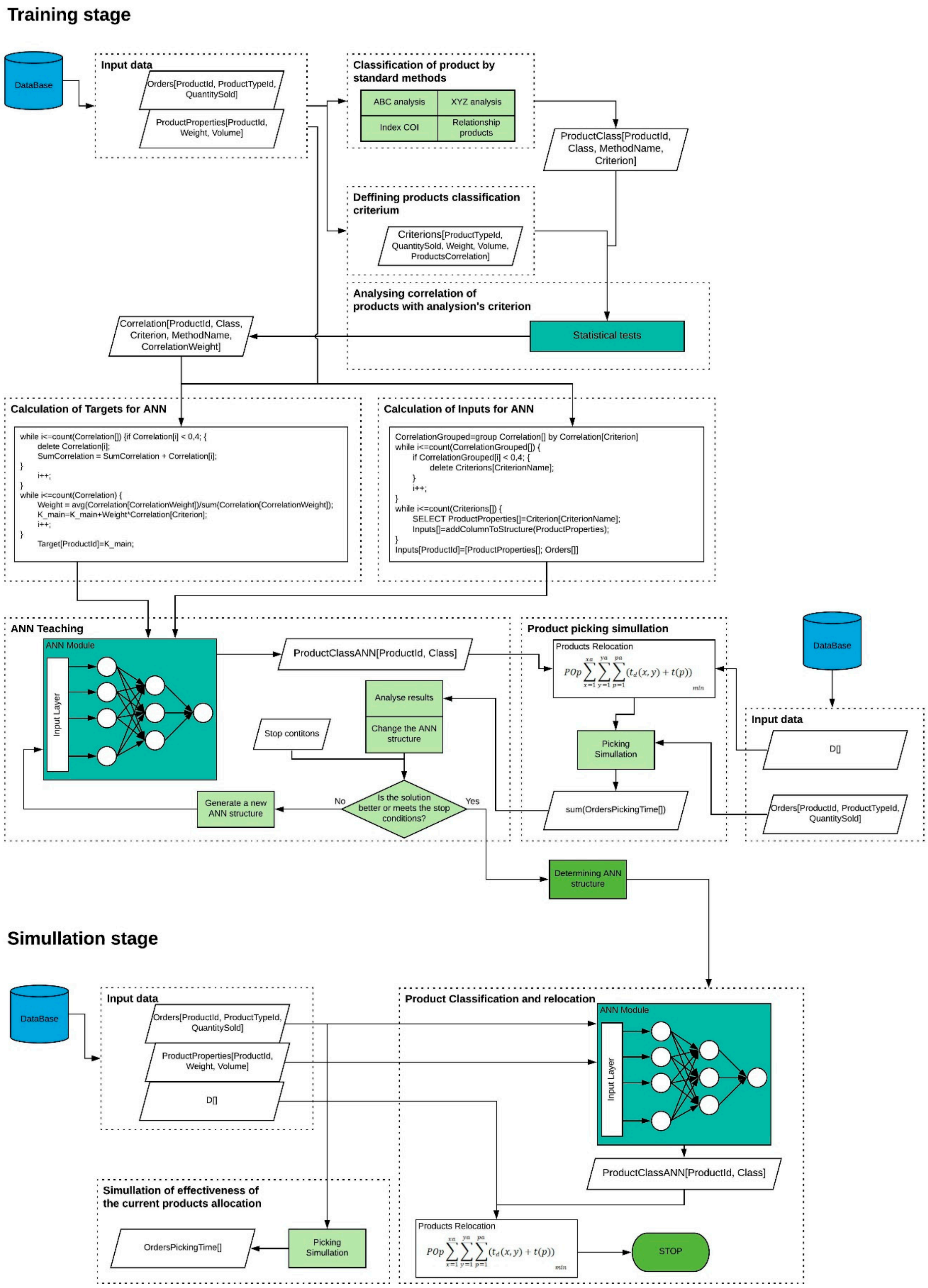
- The warehouse structure is described in a matrix form. This allows for calculating the distances between individual points in a warehouse, including warehouse parameters such as rack width (W) and rack length (L), the width of a picking aisle between storage racks, and the number of picking and cross-warehouse aisles;
- The warehouse does not employ "goods-to-man" order-picking systems, e.g., S/R (Storage/Retrieval) machines and conveyors in the product storage area. All orders are completed by the order-pickers using forklifts and order-picking trucks in the storage area. The beginning and conclusion of the order-picking process take place in the packaging area of the warehouse;
- The order-picking process can be supported by any warehouse management system (WMS) and automatic identification tools (terminals);
- The warehouse can store products with significant weight and volume differences;
- It is permissible to have a separate storage area in the warehouse for goods with different characteristics;
- The velocity profile is based on the normal distribution and the average velocity configuration ($v \neq$ const).

3.2. The Model Description

The developed model uses as the input data sales data (clients order-picking lists), stored products properties, and the warehouse layout. The input data are analyzed to find the correlation between products. Standard product classification methods (ABC analysis, XYZ analysis, Index COI), along with the authors' method for determining a relationship between products (frequency of occurrence in the clients' orders), are used.

The proposed model is presented in Figure 1 as a block diagram.

The results of the products classification are then compared with the products' parameters and the demand for the product by statistical analysis. Next, the best classification method and the most important classification criterion are used further for the automatic preparation of teaching data sets for the artificial neural network. After teaching the ANN, the simulations of product classification and allocation are executed. Furthermore, the performance analysis is accomplished, and the ANN structure is modified. The simulations, performance analysis and ANN structure modification are repeated in a sequence until the stop condition is reached (the improvement of the results is low, or the simulation time is too long). On reaching this condition, the model gives a value of the products classification and the product allocation layout.


Training stage
DataBase
Input data
Orders[ProductId, ProductTypeId, QuantitySold]
ProductProperties[ProductId, Weight, Volume]
Classification of product by standard methods
ABC analysis
XYZ analysis
Index COI
Relationship products
ProductClass[ProductId, Class, MethodName, Criterion]
Deffining products classification criterium
Criterions[ProductTypeId, QuantitySold, Weight, Volume, ProductsCorrelation]
Analysing correlation of products with analysion's criterion
Statistical tests
Correlation[ProductId, Class, Criterion, MethodName, CorrelationWeight]
Calculation of Targets for ANN
Calculation of Inputs for ANN
ANN Teaching
ANN Module
Input Layer
ProductClassANN[ProductId, Class]
Analyse results
Change the ANN structure
Stop contitons
Is the solution better or meets the stop conditions?
No
Yes
Generate a new ANN structure
Product picking simulation
Products Relocation
$POp \sum_{x=1}^{xa} \sum_{y=1}^{ya} \sum_{p=1}^{pa} (t_d(x,y) + t(p))_{min}$
Picking Simulation
sum(OrdersPickingTime[])
DataBase
Input data
D[]
Orders[ProductId, ProductTypeId, QuantitySold]
Determining ANN structure
Simullation stage
DataBase
Input data
Orders[ProductId, ProductTypeId, QuantitySold]
ProductProperties[ProductId, Weight, Volume]
D[]
Product Classification and relocation
ANN Module
Input Layer
ProductClassANN[ProductId, Class]
Simullation of effectiveness of the current products allocation
OrdersPickingTime[]
Picking Simullation
Products Relocation
$POp \sum_{x=1}^{xa} \sum_{y=1}^{ya} \sum_{p=1}^{pa} (t_d(x,y) + t(p))_{min}$
STOP

```
while i<=count(Correlation[]) {if Correlation[i] < 0,4; {
        delete Correlation[i];
        SumCorrelation = SumCorrelation + Correlation[i];
}
        i++;
}
while i<=count(Correlation) {
        Weight = avg(Correlation[CorrelationWeight])/sum(Correlation[CorrelationWeight]);
        K_main=K_main+Weight*Correlation[Criterion];
        i++;
}
        Target[ProductId]=K_main;
```

```
CorrelationGrouped=group Correlation[] by Correlation[Criterion]
while i<=count(CorrelationGrouped[]) {
        if CorrelationGrouped[i] < 0,4; {
                delete Criterions[CriterionName];
        }
        i++;
}
while i<=count(Criterions[]) {
        SELECT ProductProperties[]=Criterion[CriterionName];
        Inputs[]=addColumnToStructure(ProductProperties);
}
Inputs[ProductId]=[ProductProperties[]; Orders[]]
```

Figure 1. The block diagram of a model.

The developed model has the features of artificial intelligence. It searches for correlations between the output and the results of the "classical" analysis, which are used for establishing a set of "learning data and targets" for the learning process of ANN. According to the results obtained from artificial neural networks (ANN), the model performs the simulation automatically, evaluates the efficiency of the obtained solution, and corrects the structure of the ANN. Work done by the analyst has been minimized, e.g., the introduction of parameters describing the structure of the warehouse and the selection of classification criteria (relevant for the particular warehouse).

The main differences between the proposed model and models used in the literature so far are shown in Table 1.

Table 1. The differences between the proposed model and the standard models.

The Difference Criteria	Proposed Model	Models Presented in the Literature
Number of analyzed criteria	Analyzing four criteria at the same time	Analyzing one or just a few classification criteria [7,9]
Complexity	Model base on demand, sales statistics, can be used for large assortments, a different type of warehouse and products	Basic—method uses a simple algorithm [1,10–12]
The core of the model	Artificial intelligence—artificial neural networks, statistics, simulation of picking time	Block algorithm [2], Genetic algorithm [10]
Analyzing picking distances	Yes	No [1]
Taking velocity into account	Yes	No [1]
Taking a warehouse structure into account	Yes—for a homogeneous type of racks	No [1]
The ability to adapt to changes in demand, assortment (with parameters) and storage conditions	Yes—automatic for each of them	For demand, and just for the basic data of assortment [13–15]
Analyzing the quality of the result	Yes—automatic analysis of results is based on simulation	No, just by additional works [1]
Input data	List of products with parameters, products demand, warehouse structure, the velocity of forklifts, picking lists	List of products with basic parameters, products demand [2]
Output data	Product classification, location of products in the warehouse	Product classification [2]

[1]—the distance is not calculated for the classification method but for studies related to picking. [2]—studies related to the ABC, XYZ, COI, EIQ models.

The comparison of the proposed method to the other method presented in the research paper is in Table 2.

Table 2. Comparison of the product classification methods with the proposed model.

Method	Multicriteria	Commonly Used Criteria	Limited to a Quantity of Products Indexes	Classification Method	Most Common Uses

Index COI	Yes, just 2	volume/weight;
		products demand;
			no	Based on the COI, no grouping required	Free storage space method

Analyse ABC	No	the value of sales;
		profit from sales;
		picking frequency;
		quantity of products sales;
		volume/weight;
			low	Based on the product share of the total, products grouping	The method of permanent storage places

Analyse XYZ	No	seasonality;	low	Based on the product share of the total, products grouping	Coupled with ABC analysis

Analyse ABC-FC	No	the value of sales;
		profit from sales;
		picking frequency;
		quantity of products sales;
		volume/weight;
			low	Based on membership function and division into groups according to ABC analysis	Coupled with ABC and XYZ analysis

Analyse EIQ	Yes	clients' orders size;
		quantity of products ordered by a client;
		orders quantity by-products;
		product demand;
			low	Based on the calculated factors for each product	Method of free and permanent storage places

Method AHP	Yes	the value of sales;
		profit from sales;
		picking frequency;
		volume/weight;
		seasonality;
			high	Based on the calculated preference indicators and the adopted weighting factors	Small assortment

Proposed model	Yes	picking frequency;
		products parameters;
		volume/weight;
		demand for products;
		correlation between products;
		seasonality;
			no	Based on orders' correlation with products parameters and demand, the core of the model base on ANN	Method of free and permanent storage places, big assortment

Source: own elaboration.

3.3. Sub-Modules of the Presented Model

In continuation, the sub-modules of the proposed model will be presented and discussed.

3.3.1. Warehouse Structure Generation Module

The proposed model can be implemented in most general warehouses, i.e., those with a storage rack system for storing products on pallets. The parameters for the storage compartments have to have the same dimension due to a matrix-parametric description of the warehouse structure. Multiplying the storage compartments both in the horizontal and the vertical direction gives a storage rack with columns and tiers.

In this particular model, it is necessary to define additional parameters for the dimensions of the storage racks and picking aisles, as well as locations where the picking aisles between the storage racks are situated. The warehouse structure is described by a transition matrix, the so-called incident matrix. Therefore, the structure of the warehouse storage area can be written in the form of a ***W*** transition matrix.

$$W = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1y} \\ r_{21} & r_{22} & \cdots & r_{2y} \\ \vdots & \vdots & \ddots & \vdots \\ r_{x1} & r_{x2} & \cdots & r_{xy} \end{bmatrix} \tag{1}$$

For example, for a warehouse with eight (8) storage racks, four (4) picking aisles and one (1) cross-warehouse aisle in the middle, the warehouse structure in the form of a matrix may be presented as follows:

$$W = \begin{bmatrix} 1&1&1&1&1&1&0&1&1&1&1&1&1 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 1&1&1&1&1&1&0&1&1&1&1&1&1 \\ 1&1&1&1&1&1&0&1&1&1&1&1&1 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 1&1&1&1&1&1&0&1&1&1&1&1&1 \\ 1&1&1&1&1&1&0&1&1&1&1&1&1 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 1&1&1&1&1&1&0&1&1&1&1&1&1 \\ 1&1&1&1&1&1&0&1&1&1&1&1&1 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 1&1&1&1&1&1&0&1&1&1&1&1&1 \end{bmatrix} \tag{2}$$

The W sample matrix includes four picking aisles along the storage rack and one cross-warehouse aisle. The packing area always has the location coordinates r (0,1). This location lies outside the W matrix so that it cannot be switched with the location in the storage area. This solution is quite common in small and medium-sized warehouses. In addition to the warehouse layout reflecting the product storage area, it is also important to consider the number of storage tiers. They can be represented with a vector showing the access (travel) time to each tier in the storage rack. This (travel) time depends on the transport means (forklift truck) used to store/retrieve the products to/from the warehouse. The W matrix, together with the parameters describing the warehouse structure, fully reflects its actual parameters. Based on the matrix W, a matrix D is created representing the access (travel) times from the packing location r (0,1) to each single (selected) tier (storage compartment) of the storage rack.

The matrix D contains values with access times for pallets' places in racks, and a "0" value for the locations of aisles. For example, for 12 matrix rows (8 of them are representing racks rows), 12 racks in each, and one cross-warehouse aisle, the D matrix for the first level of storage will be as follows:

$$D = \begin{bmatrix} 6,6 & 8,4 & 10,2 & 12,0 & 13,8 & 15,6 & 0 & 19,2 & 21,0 & 22,8 & 24,6 & 26,4 & 28,2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 6,6 & 8,4 & 10,2 & 12,0 & 13,8 & 15,6 & 0 & 19,2 & 21,0 & 22,8 & 24,6 & 26,4 & 28,2 \\ 17,8 & 19,6 & 21,4 & 23,2 & 25,0 & 26,8 & 0 & 30,4 & 32,2 & 34,0 & 35,8 & 37,6 & 39,4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 17,8 & 19,6 & 21,4 & 23,2 & 25,0 & 26,8 & 0 & 30,4 & 32,2 & 34,0 & 35,8 & 37,6 & 39,4 \\ 29,0 & 30,8 & 32,6 & 34,4 & 36,2 & 38,0 & 0 & 41,6 & 43,4 & 45,2 & 47,0 & 48,8 & 50,6 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 29,0 & 30,8 & 32,6 & 34,4 & 36,2 & 38,0 & 0 & 41,6 & 43,4 & 45,2 & 47,0 & 48,8 & 50,6 \\ 40,2 & 42,0 & 43,8 & 45,6 & 47,4 & 49,2 & 0 & 52,8 & 54,6 & 56,4 & 58,2 & 60,0 & 61,8 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 40,2 & 42,0 & 43,8 & 45,6 & 47,4 & 49,2 & 0 & 52,8 & 54,6 & 56,4 & 58,2 & 60,0 & 61,8 \end{bmatrix} \quad (3)$$

Relying on the determined matrix D, it is possible to order individual pallets according to their importance by sorting them according to access times. The racks with the shortest access time will be able to store major goods. The lifting time is defined by the type of forklift. This time is determined by the vector. The value in the vector represents the time of picking products from the exact tier of the storage rack. The example of this vector for a high-level order-picker forklift is t(p) = [0; 58; 116; 174; 232; 290; 348; 406].

3.3.2. ANN Input Data Choosing Module

The input data module allows the analysis of a statistical relationship between input data and order-picking times. In the case of expressive dependencies, this module determines (see Table 3) the weights, adjusting the results of product classification for each model. For each pair (input parameter and product classification method), the module checks the value of the coefficient p, which, for all possible pairs, is less than 0.05 (it is a statistical linear correlation between the values in the pair at the 5% significance level). The correlation coefficient is next calculated on a scale from −1 (strong inversely proportional correlation) to +1 (strong directly proportional correlation).

Table 3. Correlation coefficients of input parameters with the analyses results.

	COI Index	ABC (Weight)	ABC (Volume)	Algorithm of Relationships Search	Sum
Number of products	0.7384	0.7724	0.7723	0.6891	2.9722
Number of product types	0.7982	0.8375	0.8342	0.7433	3.2132
Products volume (m^3)	0.6250	0.6735	0.6514	0.5594	2.5093
Products weight (kg)	0.7139	0.7512	0.7319	0.6548	2.8518
Average	0.7189	0.7587	0.7475	0.6617	2.8866

Source: own elaboration (Statgraphics Centurion 15.2.11.0).

The graphs showing the relationships for individual pairs are shown in Figure 2. The correlation coefficients are shown in Table 3.

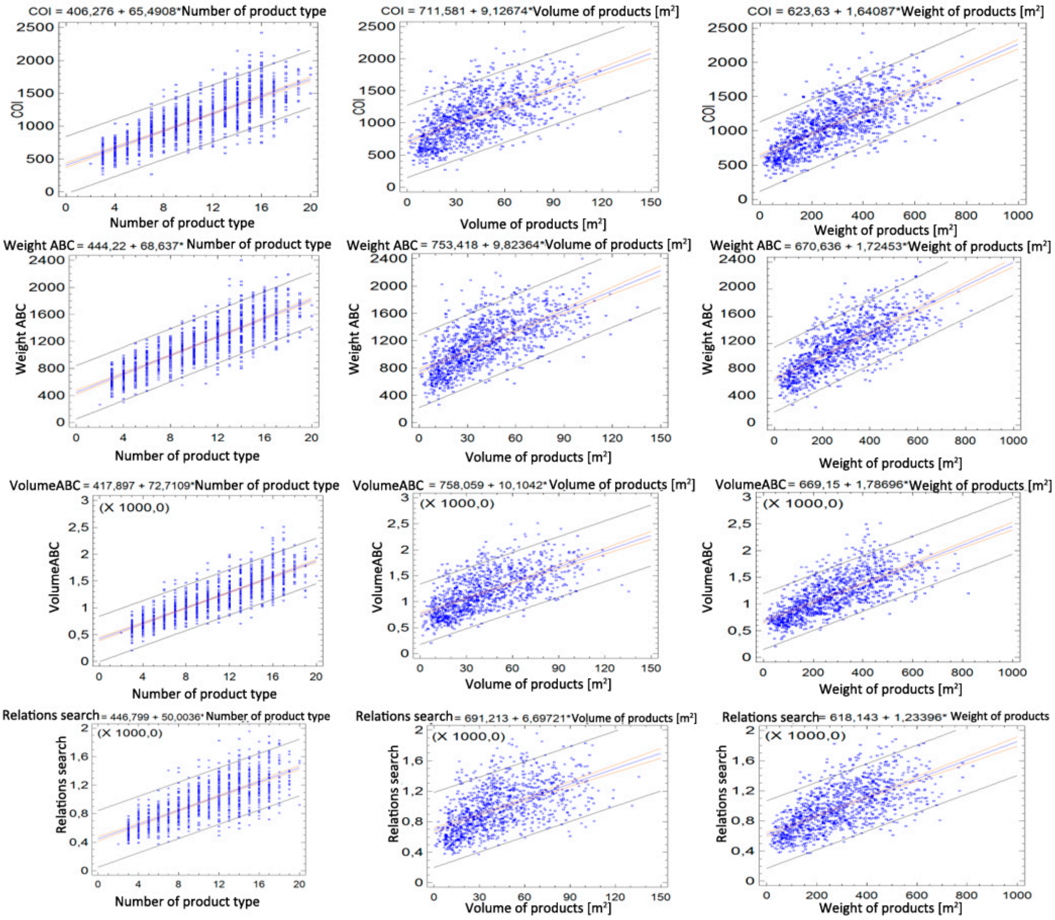


Figure 2. Relationships between order-picking time and input parameters for individual analysis, source: own elaboration.

3.3.3. ANN Target Selection Module

Weight coefficients (w) are based on the calculation of the average categories (k), returned as results from the different product classification (n) methods (4):

$$w = \frac{\bar{k}}{\sum_{i=1}^{n} \bar{k}} \tag{4}$$

For the selected case study, the module has obtained the following weighting factors (5):

$$w = \begin{cases} 0.2490 & for\ COI\ Index \\ 0.2628 & for\ ABC\ analysis\ (weight) \\ 0.2589 & for\ ABC\ analysis\ (volume) \\ 0.2292 & for\ Method\ of\ relationships\ search \end{cases} \tag{5}$$

The final category (k_{main}) employed in the learning process of ANN is described in expression (6):

$$k_{main} = 0.2490 \cdot kat_{COI} + 0.2628 \cdot kat_{ABC} + 0.2589 \cdot kat_{ABC2} + 0.2292 \cdot kat_{AP} \tag{6}$$

3.3.4. ANN Learning Module

The developed system allows integration with the MatLab software to model ANN. MatLab provides many algorithms for ANN training. Applying different learning algorithms results in different outcomes [27], some of which are used in this publication. As a function of the objective, minimization of the mean square error (MSE) was assumed. The consequential floating-point results were mathematically rounded off to the integers corresponding to the class of product affiliation. Afterwards, statistical analyses were performed to determine which of the methods allows one to designate a category of products that would allow for shorter order-picking times. Among the available learning algorithms, the most popular ones were used, e.g., [28]:

- Trainoss—One step secant backpropagation;
- Traingd—Gradient descent backpropagation;
- Trains—Sequential order incremental training with learning functions;
- Trainlm—Levenberg–Marquardt backpropagation.

To select the best learning algorithm, the simulations for the same model with different learning algorithms were done. The learning algorithms were compared by the quality of the simulation result. The criterion for selecting this algorithm was the minimization of the order-picking time. The achieved results are presented in Figure 3.

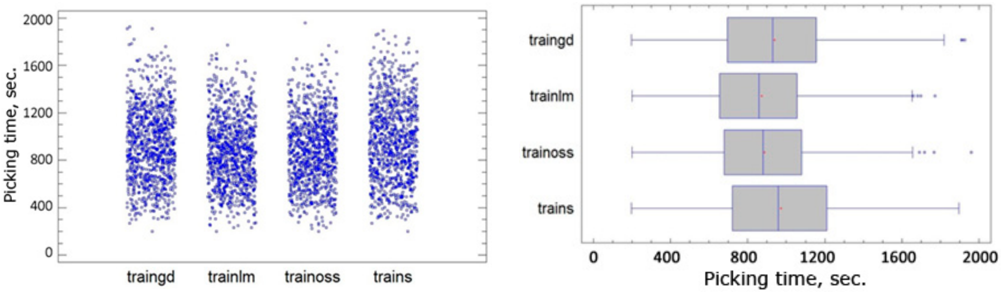


Figure 3. Time of product picking depending on the ANN learning function used, sec.; on **left**—results for each case as a point cloud, on **right**—results as box-and-whisker chart; source: own elaboration (Statgraphics Centurion 15.2.11.0).

As can be seen in Figure 3, the most different results, with comparison to others, are obtained after applying the trains learning function—a sequential method of modifying weights with a learning function. By making a preliminary review of statistical data, it can be concluded that the *trainlm* function can achieve the best result since it has the lowest average value and median, while maintaining the smallest standard deviation, scope and maximum duration of order-picking (Table 4).

Table 4. Statistics of order-picking time depending on the ANN learning function, sec.

Learning Function	Average	Standard Deviation	Median	Coefficient of Variation	Min.	Max.	Range
traingd	938.3	312.8	929.7	0.333	198.4	1926.4	1728.0
trainlm	872.1	288.9	857.5	0.331	201.2	1771.2	1570.0
trainoss	886.7	292.1	879.2	0.329	201.2	1958.0	1756.8
trains	971.5	334.3	958.2	0.344	198.4	1894.0	1695.6

Source: own elaboration (Statgraphics Centurion 15.2.11.0).

For a selected case study, the following relationship of training data sets was made:

- 70% of the collection—training data;
- 15% of the collection—validation data;
- 15% of the collection—testing data.

During data preparation, the following Matlab functions were applied: Mapminmax, Removeconstantrows.

The *Mapminmax* function is used to normalize the input data, which also reduces the necessary calculation time. Before training, the inputs and targets are scaled so that they always fall within a specified range. The function *mapminmax* scales inputs and targets so that they fall in the range [−1,1].

The original network inputs and targets are given in the matrices *p* and *t*. The normalized inputs and targets *pn* and *tn* contain the settings; in this case, the minimum and maximum values of the original inputs and targets. After the network has been trained, the ps settings should be used to transform any future inputs that are applied to the network. They effectively become a part of the network, just like the network weights and biases.

If *mapminmax* is used to scale the targets, then the output of the network will be trained to produce outputs in the range [−1,1]. To convert these outputs back into the same units that were used for the original targets, the settings *ts* have been used.

The network output corresponds to the normalized *targets tn*. The unnormalized network output is in the same units as the original targets *t*.

If *mapminmax* is used to preprocess the training set data, then whenever the trained network is used with new inputs they should be preprocessed with the minimums and maximums that were computed for the training set stored in the settings *ps*. The code (12–14) applies a new set of inputs to the already-trained network.

For most ANNs in the MatLab software, including *feedforwardnet*, the described steps are done automatically, so there is only one need to use the sim command.

It is assumed that (X) has only finite real values, and that the elements of each row are not all equal. If *x_max* = *x_min* or if either *x_max* or *x_min* is non-finite, then *y* = *x*, and no change occurs.

The *Removeconstantrows* function is responsible for removing inputs consisting of equal values (16). Any NaN (Not a Number) values in the input matrix are treated as missing data and are not considered as unique values. So, for example, *removeconstantrows* removes the first row of the matrix.

3.3.5. ANN Simulation Module

The ANN structure has been selected based on the method of successive approximation. The different structures (different number of layers and number of neurons in it) were examined and the next structure was determined, based on the results from the simulation. Then results from both simulations were compared to propose another structure. The selection of the network structure was made by assessing the mean square error (MSE).

For the network training the following parameters were assumed:

1. Epochs: 1000;
2. Time: Inf;
3. Goal: 0;
4. Min_grad: $1 \times 10^{-5;}$
5. Max_fail: 6;
6. Mu: 0.001;
7. Mu_dec: 0.1;
8. Mu_inc: 10;
9. Mu_max: 10,000,000,000.

The model finds the optimal structure of the network—self-learning. This self-learning is done by a change of the neuron's weight in the ANN, and it simulates the performance of order-picking. If the quality of the model after changing the structure is better, this structure is chosen. After this, the change in the weight of ANN and the performance analysis is done again. This loop is done until the stop condition is achieved (number of loops). The final structure of the ANN was chosen for the network of high quality—the results from the simulation are similar to the real data. The best results have been obtained by a multilayer unidirectional feedforward type network consisting of three hidden layers. A simplified diagram of the applied neural network is shown in Figure 4. Figure 5 shows an example of a graph of a mean square error that occurred during the process of learning for a selected network structure and for an exemplary structure that was not considered optimal.

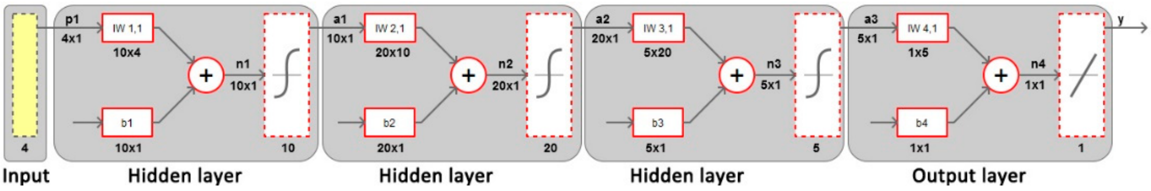


Figure 4. A structure of the applied ANN, source: own study.

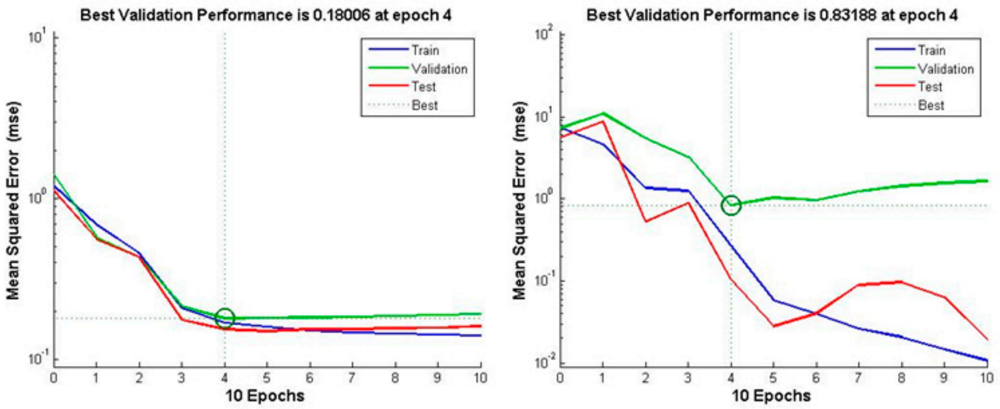


Figure 5. SNN training results, mean-square error for chosen structure (**on the left**), and rejected structure (**on the right**); source: own elaboration (MatLab R2011b).

Feedforward networks are characterized by the specific direction of signal flow—it is for the specified input to output on which the network provides an established solution. *Feedforward* networks are both the most commonly used and most useful. In the first hidden layer, 10 neurons were used, in the second, 20, and in the third, 5. In all hidden layers, the tangential activation function was applied. The input data used were the number of units sold per year, the product's popularity, its weight, and the volume.

For the selected network structure, a histogram of network learning errors has been generated and shown in Figure 6.

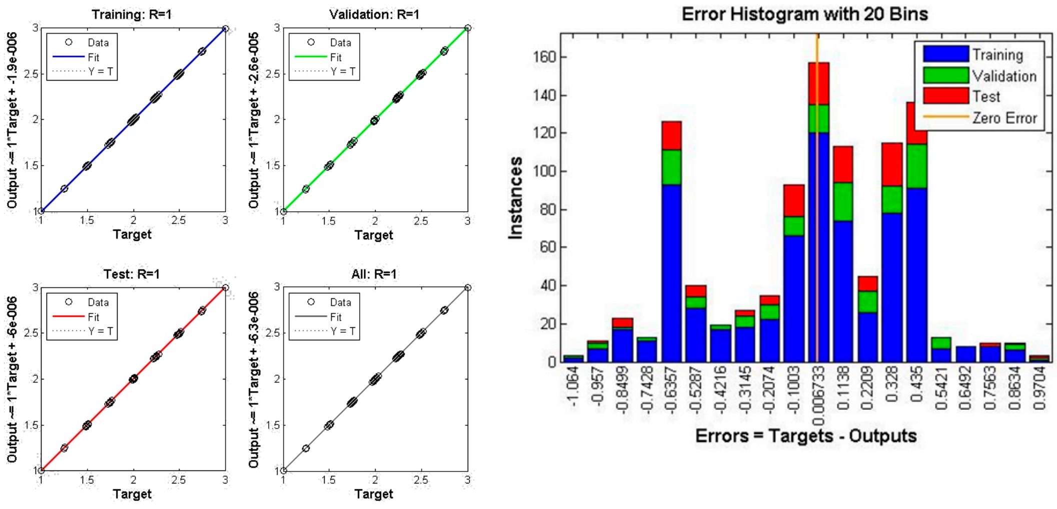


Figure 6. The regression and histogram of network learning errors; source: own elaboration (MatLab R2011b).

The developed ANN structure gives high-quality results—similar to the target values (Figure 6, on left). It can be noticed that the output values from the training ANN are similar to the target value. This proves that the developed model enables us to achieve very similar results to the real ones.

3.3.6. Product Classification and Allocation Module

Another module is the classification and simulation one, consisting of three steps: product classification, placement planning and order-picking simulation. In the product classification step, the user can select one of the classification methods, such as the association search algorithm, ABC analysis combined with XYZ analysis, or solo ABC analysis, as well as a model linking them to the COI Index and the author's algorithm for the relationships search. For the last two methods, a user should also specify the criterion of analysis—product popularity, number of units sold, weight or volume. In the simulation settings, a user only has to select the product classification model used together with the criterion and then specify the number of pick lists to be included in the simulation. Other simulation settings are defined in the program's initial settings—the ANN structure generation module and core system settings—so a user does not need to re-enter them. Once the selection is confirmed, the program simulates successively calculating the time to go to the rack and retrieve a specified number of items from it. The result of the simulation is the order-picking time, the penalty resulting from the need to change the order of products on the pick-up truck (additional time) and the total order-picking time with the penalty. In addition, the program provides statistical picking data, from the average stock level through the total number of items in the order, and warehouseman movement speed up to a number of items in the list, as well as total weight and volume.

ABC analysis is the most commonly used analysis to classify products. Classic ABC analysis allows for dividing items into three groups with a percentage share of A—80%, B—15%, C—5%. Variations of this analysis can be found. They isolate more groups in a properly adjusted percentage share. ABC analysis is a one-criterion analysis so it is not possible to take several input parameters simultaneously into account. However, it is possible to do several analyses, taking another as a criterion feature, and then to synthesize the results by taking the appropriate weight for each criterion (analysis result) [29]. The mentioned analysis is most often performed according to the criteria [30–32]:

- sales value or gain from sales;
- the frequency of picking;
- delivery size;
- weight and volume.

ABC analysis is often compared with the Pareto–Lorenz classification, whereby class A, consisting of 20% of all items, generates 80% of the total value of the set.

XYZ analysis allows for supplementing the ABC analysis with an additional criterion by making a classification within already isolated groups. The criterion most commonly used in XYZ analysis is the regularity of consumption based on historical sales data. Unlike ABC analysis, where the popularity of the product or the number of units sold is analyzed, the XYZ analysis allows independent evaluation based on an individual factor for each product. The basic assumptions are the same for both of the aforementioned analyses, e.g., the period used for the calculation must be common to all products and the compliance of the period and measuring units must be ensured. XYZ analysis is performed in the following steps:

Step 1—A compilation of data from equal periods. However, the greater the number of periods, the better the results will be;

Step 2—A calculation of the average number of units sold (7) for each product:

$$\overline{w_t} = \frac{\sum_{j=1}^{n} w_{ij}}{n} \tag{7}$$

Step 3—A calculation of standard deviation (8):

$$s_i \ = \ \sqrt{\frac{\sum_{j=1}^{n} (w_{ij} - \overline{w}_t)^2}{n-1}} \tag{8}$$

Step 4—A calculation of the coefficient of variation (9) for each product:

$$V_i = \frac{s_i}{\overline{w_t}} \tag{9}$$

Step 5—Sorting out the product list in ascending order by the value of the coefficient of variation V_i;

Step 6—Defining limit values for each group. Usually, the values of the share are $X = 0.2V_i$, $Y = 0.4V_i$;

Step 7—Allocation of products to groups: $X \in (0; 0.2]; Y \in (0.2; 0.4]; Z \in (0.4; 1]$.

As a result of the analysis, the products are divided into XYZ groups, and it is assumed that [12]:

- The X group consists of products that are used regularly. It should have the monitoring system which is synchronized with the demand. It is not required to create large stocks of goods;
- The Y group is characterized by considerable variability, which makes it difficult to see and use certain patterns, such as seasonality. It is necessary to develop an appropriate level of stockholding;
- The Z group consists of occasional products. It is necessary to choose between creating a stock that will be stored for a long period and on-demand supplies that will require the incurrence of higher expenses.

The developed system allows for combining the results provided by XYZ analysis with ABC analysis.

The COI Index (cube-per-order index) allows classifying products in the simplest way [13]. This model provides bi-criterion analysis, and the criteria are product size and the demand. The size of the product can be defined by its volume or weight, while the demand by the number of products picked (popularity) or the average demand. The COI Index is calculated as follows (10):

$$w_i \ = \ \frac{v_i}{p_i}; \ i \ = \ 1, 2, \ldots, n \tag{10}$$

By applying the calculated COI Index to distribute products so that the items with the lowest factor are as close to the packaging/delivery area as possible, the path travelled by the largest/heaviest products is shortened. Making a number of goods conditional on the demand allows for striking a balance between these two criteria [30].

The additional advantage of this model is that frequent changes in the order of products on the trolley are avoided, while completing the products within a large weight range and planning the picking process of several orders in one pass from the farthest point to the release zone. Such changes are necessary if there is a risk of damaging the lighter product with the heavier one. Employing the COI Index model indirectly reduces the time required for the product to be transferred, and as a result, reduces the duration of the complete picking. It is possible to use the classic ABC model by double-taking it—once with the weight criterion, and again with the demand for the product criterion. The next step is to determine the weighting factors to compile the results of both analyses. However, with a large assortment, the use of the COI Index will be a much faster method.

The COI Index is particularly useful for the free allocation storage method. In most cases, it has an informational and decision-making function and contributes to the reduction of picking time without the need for complex calculations or analyses. The literature on the subject provides information on the use of the COI Indexes to divide products into ABC groups [33]. For this reason, the COI Index model happens to be wrongly associated with ABC analysis.

The search algorithm for relationships between products. One of the authors' developed models is the algorithm for determining correlations between products. This is automatic. The application generates a list of pickups according to the configuration settings of the simulation. Consequently, using the author's algorithm, it calculates the number of product appearances in pairs on the pick lists and writes them to the matrix. The last step is to generate a set of potential pick lists with different product appearances.

When classifying products, the relationships between them, e.g., the frequency of occurrence on the picking lists, can be used as a criterion for allocation. In such a case, the placement of products is based on the list of products sorted by the greatest number of appearances on the picking lists. Such a solution is not entirely correct as it does not reflect the relationships between the individual products, focusing only on the frequency of their sales. Every link should be contained in the matrix of relationships to take into account the relationships between products. The main problem is the arrangement of products, such that the products that more frequently occurred together on the lists were allocated close to each other. To solve this, a model has been developed to organize the products according to the previously mentioned matrix of relationships.

The methodology for sorting products on the picking lists according to the links between them can be presented in the following steps:

Step 1—Creating the matrix A, the matrix of frequencies occurring on the picking lists (11)

$$A = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{x1} \\ w_{21} & w_{22} & \cdots & w_{x2} \\ \vdots & \vdots & \ddots & \vdots \\ w_{1y} & w_{2y} & \cdots & w_{xy} \end{bmatrix} \tag{11}$$

Step 2—Counting the sum s_u (12) of rows of the matrix A

$$s_u = \sum_{idx=1}^{i} w(idx, idy) \tag{12}$$

Step 3—Entering the sum s_u to matrix S (13)

$$S = \begin{bmatrix} id_{prod1} & id_{prod2} & \cdots & id_{prody} \\ s_{u1} & s_{u2} & \cdots & s_{uy} \end{bmatrix} \tag{13}$$

Step 4—Determination of the identifier (*ident*) for the element of the highest value in matrix S. If several elements have the same maximum value, then all identifiers must be read;

Step 5—Rewriting element values from matrix A to the new matrix B for predetermined identifiers (*ident*);

Step 6—Assigning the identifier to vector W that stores IDs for products arranged by several links in decreasing order;

Step 7—Reading product IDs for identifiers *idx* and *idy* of the element of the highest value in matrix B;

Step 8—If several elements have the same maximum value, then IDs should be read for the value which corresponds to the element of the highest value in matrix S;

Step 9—Verification of whether *idx* or *idy* are in vector W;

If they are not, both values *idx* and *idy* must be added to the vector W before the next step.

It they are, proceed to the next step;

Step 10—Replacing the current value in matrix B with a value 0 for the maximum selected in step 7;

Step 11—Adding values from matrix A to the B element of the identifier (*ident*) that was the last one added. Checking the condition (14),

$$\max(B) = 0 \tag{14}$$

If it is fulfilled, then the algorithm can be terminated.

If it is not fulfilled, then proceed to the next step;

Step 12—Verification of whether the number of values in vector W is equal to the number of analyzed products.

If it is, then the algorithm can be terminated.

If it is not, go back to step 7.

The product allocation in the warehouse was based on the designation of the location characterized by the shortest access time from the packaging zone described by the formula (15).

$$POp\sum_{x=1}^{xa}\sum_{y=1}^{ya}\sum_{p=1}^{pa}(t_d(x,y)+t(p))_{min} \tag{15}$$

3.3.7. Efficiency Analysis Module

The assessment of the storage processes' efficiency can be performed in terms of different criteria, including (a) the duration of picking, (b) the route of picking, and (c) resource consumption (human labor, costs, transport). The developed system as a criterion of efficiency adopts the duration of order-picking.

To evaluate the effectiveness of the picking process that results from the applied model of product distribution in the warehouse, it is important to apply a model that allows high precision so as to reflect the processes occurring in a real warehouse.

According to the picking lists, a picking route was planned using a model applied in the majority of medium-sized companies that have their warehouses, i.e., determining the order-picking route based on the closest point to the current location of a warehouse employee. This model also contemplates the need to intervene in the order of products on a picking table in the event of the risk that a smaller product can be crushed by a larger one—sensitivity to stacking.

The model can be implemented in most of the conventional warehouses, i.e., those where the products are stored in rows in the storage area. There can be many parameters of the racks; however, all the racks must be the same, which is important because of a matrix-parametric description of warehouse structure. The number of storage levels is irrelevant; the warehouse may have several levels of storage (Lorenc and Lerher, 2019). Incidence matrix was used to describe the structure. Accordingly, the structure of the warehouse storage surface can be expressed as an incidence matrix *W* (16).

$$W=\begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1y} & \cdots & r_{1M} \\ r_{21} & r_{22} & \cdots & r_{2y} & \cdots & r_{2M} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ r_{x1} & r_{x2} & \cdots & r_{xy} & \cdots & r_{xM} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{N1} & r_{N2} & \cdots & r_{Ny} & \cdots & r_{NM} \end{bmatrix} \tag{16}$$

For the precision of further considerations, the following notation is used $r_{xy} \equiv r(x,y)$. The *W* matrix along with the parameters describing the structure of the warehouse fully reflects its real dimensions. Based on the *W* matrix, a *D* matrix was constructed that mirrors access time from packaging area *r*(0,1) to each rack located in the storage area.

Before completing the values in matrix *D* it is necessary to fill in the ***W*** matrix.

The movement time from the packaging area to the first product on the picking list has been described by formula (17).

$$\begin{gathered} t(x,y,p) \;=\; t_{lp}\;(D_{lr}(y)\cdot(x-c_l(x)) \;+\; d_{cr}\cdot c_l(x)) \\ +t_{lp}\;(D_{wr}(y)\cdot(y-c_w(y)) \;+\; d_{cr}\cdot c_w(y)) \;+\; t_a \;+\; t(p) \end{gathered} \tag{17}$$

For further calculations, in the paper, the time of lifting and lowering forks *t(p)* on a given storage level was assumed based on the technical data of a standard high storage forklift truck.

Besides, this requires a time correction t_p if the rack is facing the packaging zone, because it will not be necessary to travel an extra distance equal to the width of the corridor between the rows. Therefore, the correction takes the forms of equation (18).

$$If\ W\ (x, y-1) \ = \ 0\ then\ t(x, y, p)_{new} \ = \ t\ (x, y, p) - \ t_{lp}\ (D_{wr} + d_{cr}) \tag{18}$$

After completing the entire D matrix with the access times to a given spot, it is important to exclude the values with access times to spots that are not storage spots (are aisles). Racks with the shortest access time will be used to store key goods.

To analyze the total picking time, it is important to determine each time the movement time around the warehouse for each item on the picking list. Due to a large number of combinations, it is impossible to generate a matrix containing these times. In such a case it would be necessary to generate as many matrices as there are coordinates defining the location of a product.

The subsequent part of the algorithm for defining picking time can be described in the following steps.

Step 1—In the first step of calculating picking time, an absolute value is calculated, from a difference between the initial starting point and the ending one (19) for racks and rows.

$$\begin{aligned} x_{temp} &= \left|x_{pocz} - x_{konc}\right| \\ y_{temp} &= \left|y_{pocz} - y_{konc}\right| \end{aligned} \tag{19}$$

Step 2—In the second step, it is verified whether the racks are in the same row, and whether or not it is the row coming out directly from the packaging area—the starting point.

$$y_{temp} \ \neq \ 0\ or\ y_{temp} \ \neq \ 2 \tag{20}$$

Step 3—If the condition of step 2 was fulfilled, then in the calculation of the time it is important to take two curves into account, $l_{luk} = 2$. Besides, as in the contemplated warehouse, where two-row racks with constant access provided are used (every third row is a corridor between the racks), it is necessary to contemplate several longitudinal corridors (21)

$$l_{korwz} \ = \ \frac{y_{temp}}{3} \tag{21}$$

In this case, the length of a road across is described by formula (22).

$$dr_{poprz} \ = \ D_{wr}(y) \left(y_{temp} - l_{korwz}\right) + \ d_{cr} \cdot l_{korwz} \tag{22}$$

Step 4—The analyzed warehouse has one transversal corridor; therefore, before calculating the road along the warehouse, it is important to verify whether the two points are in the same or different sections of the warehouse, meaning they are separated by a corridor (23).

$$\left(x_{pocz} \ \leq \ \frac{l_{reg}(y)}{2} oraz x_{konc} \ > \ \frac{l_{reg}(y)}{2}\right) or \left(x_{konc} \ \leq \ \frac{l_{reg}(y)}{2} oraz x_{pocz} \ > \ \frac{l_{reg}(y)}{2}\right) \tag{23}$$

Step 5—If the condition in step 4 is satisfied, then the road along the warehouse (d_{rwzd}) should be calculated from the formula (24).

$$dr_{wzd} \ = \ D_{lr}(y) \cdot d_{cr}\left(\left|x_{pocz} - x_{konc}\right| - 1\right) \tag{24}$$

Step 6—If the condition in step 4 is not satisfied, then it is important to verify in which part of the warehouse (before or after the transversal corridor) both considered points are located (25).

$$x_{pocz} < \frac{l_{reg}(y)}{2} \tag{25}$$

Step 7—If the condition in step 6 is satisfied, then both points are located in the first area of the warehouse, so if

$$D_{lr} \cdot d_{cr}\left(x_{pocz} + x_{konc}\right) < D_{lr} \cdot d_{cr}\left(l_{reg} - x_{pocz} - x_{konc}\right) \tag{26}$$

then

$$dr_{wzd} = D_{lr} \cdot d_{cr}\left(x_{pocz} + x_{konc}\right) \tag{27}$$

otherwise

$$dr_{wzd} = D_{lr} \cdot d_{cr} \left(l_{reg} - x_{pocz} - x_{konc}\right) \tag{28}$$

Step 8—If the condition in step 6 is not fulfilled then both points are located in a second zone of the warehouse, so if

$$D_{lr} \cdot d_{cr}\left(x_{pocz} + x_{konc} - l_{reg}\right) < D_{lr} \cdot d_{cr}\left(2l_{reg} - x_{pocz} - x_{konc}\right) \tag{29}$$

then

$$dr_{wzd} = D_{lr} \cdot d_{cr}\left(x_{pocz} + x_{konc} - l_{reg}\right) \tag{30}$$

otherwise

$$dr_{wzd} = D_{lr} \cdot d_{cr}\left(2l_{reg} - x_{pocz} - x_{konc}\right) \tag{31}$$

Step 9—If the condition in step 2 was not satisfied then the calculation of time should not take curves into account, $l_{luk} = 0$, nor the length of the road in the transverse corridor $dr_{poprz} = 0$. Therefore, the length of the road in the longitudinal corridor depends on whether both points are located in this part of the warehouse, or if they are separated by a corridor (32). So, if

$$\left(x_{pocz} \leq \frac{l_{reg}}{2} \; and \; x_{konc} > \frac{l_{reg}}{2}\right) or \left(x_{konc} \leq \frac{l_{reg}}{2} \; and \; x_{pocz} > \frac{l_{reg}}{2}\right) \tag{32}$$

then the points are located in different parts of the warehouse, so

$$dr_{wzd} = D_{lr} \cdot d_{cr} \left(x_{konc} - 1\right) \tag{33}$$

otherwise

$$dr_{wzd} = D_{lr} \cdot x_{konc} \tag{34}$$

Step 10—Based on the calculated picking route, it is important to calculate its duration based on formula (35).

$$t_m = t_{lp}\left(dr_{wzd} + dr_{poprz}\right) + t_a \cdot l_{luk} + t_p \tag{35}$$

4. Implementation of the Presented Model as Simulation Software

The proposed model was used as a base algorithm for implementation in the authors' own software called Pickup Simulo. The system was coded in PHP language and MySQL relational databases, and has been tested for code integrity and standards compliance. Besides, each of the developed algorithms were tested on examples, which were then manually calculated to detect logical and algorithmic errors. These types of errors would not be impossible to detect with only code validation. After a thorough examination of many different cases, errors in the application were excluded. Thus, the impact of the application on simulation results was excluded, which was extremely important from a research point of view. The design of algorithms and coding took ten months. Another four were dedicated to the validation of the application and the improvement of the identified errors.

Thanks to the proposed model it is possible to use large batches of data, clearly display the results and easily modify input parameters. The developed application allows its integration with MatLab.

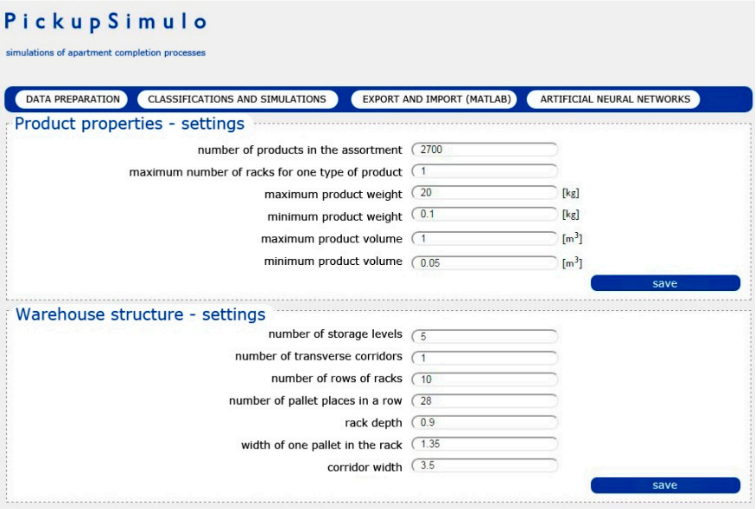
The developed system based on the presented model enables the following:

- presentation of the warehouse structure in a mathematical form;
- description of parameters of products stored in the warehouse;
- classification of products using the methods ABC, XYZ analysis and COI Index, and such criteria as product popularity, number of items sold, weight or volume;
- simulation of the allocation of products in the warehouse;
- creating a sample order-picking list;
- simulation of the order-picking process.

The application allows the performing of any number of simulations so that you can get a large set of results. In addition, the information provided is a good set of research data, which consist of the following:

- order-picking lists associated with product parameters (weight, volume, popularity, number of shelf units occupied);
- recording the distribution of products in the warehouse;
- accessing time for each product in the storage rack;
- order-picking time;
- statistical information for each order-picking, such as the order of products taken, the total volume and weight of products, the number of assortment items on the picking list and the number of pieces taken.

The first application module is a data preparation module that allows configuring the shelves, in terms of rack dimensions (Figure 7), as well as the warehouse structure (Figure 8). After entering data in the system, one can proceed to the next step of creating the warehouse structure. This part of the module is automated, based on the warehouse configuration. The program itself creates a matrix (W) corresponding to the storage locations (shelves) and corridors. In this step, the access times for the individual storage areas in the warehouse are calculated.


PickupSimulo
simulations of apartment completion processes
DATA PREPARATION
CLASSIFICATIONS AND SIMULATIONS
EXPORT AND IMPORT (MATLAB)
ARTIFICIAL NEURAL NETWORKS
save
save

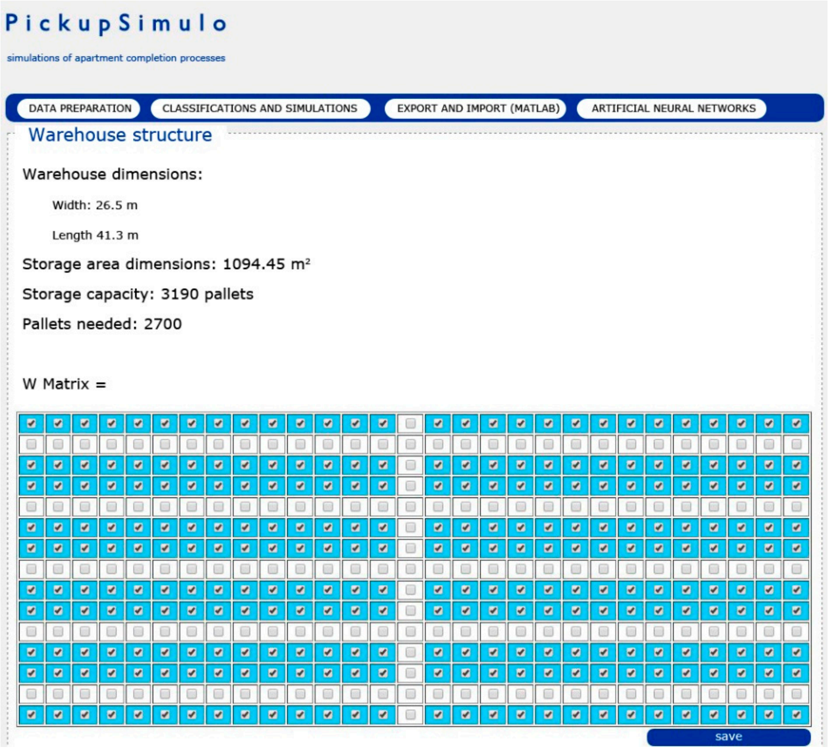
Product properties - settings

number of products in the assortment	2700	
maximum number of racks for one type of product	1	
maximum product weight	20	[kg]
minimum product weight	0.1	[kg]
maximum product volume	1	[m^3]
minimum product volume	0.05	[m^3]

Warehouse structure - settings

number of storage levels	5
number of transverse corridors	1
number of rows of racks	10
number of pallet places in a row	28
rack depth	0.9
width of one pallet in the rack	1.35
corridor width	3.5

Figure 7. Pickup Simulo, shelves' rack dimensions configuration screen.



PickupSimulo
simulations of apartment completion processes
DATA PREPARATION
CLASSIFICATIONS AND SIMULATIONS
EXPORT AND IMPORT (MATLAB)
ARTIFICIAL NEURAL NETWORKS

Warehouse structure

Warehouse dimensions:

Width: 26.5 m

Length 41.3 m

Storage area dimensions: 1094.45 m^2

Storage capacity: 3190 pallets

Pallets needed: 2700

W Matrix =

☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☐	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☐	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☐	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☐	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☐	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☐	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☐	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☐	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☐	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☐	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑

Figure 8. Pickup Simulo, warehouse structure configuration screen.

The next module is the classification and simulation module, consisting of three stages—the classification of products, the planning of their allocation in the warehouse and the simulation of order-picking.

At the stage of product classification, one of the classification methods should be selected from the following:

- the association search algorithm;
- ABC analysis combined with XYZ analysis;
- ABC analysis, modification of ABC analysis with dynamic boundary-setting combined with COI Index.

For the last two methods, an analysis criterion should be specified—product popularity, number of items sold, as well as weight or volume. In the case of the association search algorithm and XYZ analysis, the criteria are determined automatically. Moreover, for the ABC analysis, it is possible to additionally take the COI Index into account.

The last option of this module is to simulate the order-picking. The first step is to set the simulation parameters. After confirming the selection, the program performs a simulation, successively calculating the time of going to the rack and picking a certain number of products from it. The order of product picking is determined on an ongoing basis by designating the storage place of the product on the picking list that is closest to the current warehouse position. The result of the simulation is the order list picking time. The third module of the application is the module of data exchange between the described system and MatLab. The program prepares the data needed for the artificial neural network teaching process, i.e., product properties (mass, volume, number of items sold, number of connections between the product and others), and the desired product classes calculated in a manner relying on weights. The prepared data are saved in the form of a csv file, which is then loaded into MatLab. After performing calculations in Toolbox Neural Network and saving the result table to a csv file, it is possible to load the results of product classification into the developed program by using artificial neural networks.

The application allows the performing of any number of simulations, thanks to which it is possible to obtain a large set of results. Moreover, the obtained information constitutes a good set of research data, which consist of:

- order-picking lists related to product parameters (weight, volume, popularity, number of occupied racks);
- plan for the allocation of products in the warehouse;
- access time from the packing;
- dispatch area to a product on the rack;
- order-picking time;
- statistical information for each picking, such as the order of picked products, the total volume and weight of products, the number of assortment items on the picking list and the number of picked items.

The above data are sufficient to perform the simulations needed to achieve the planned research objectives.

5. Case Study

The proposed model implemented in the software Pickup Simulo was used to perform simulations for five warehouses of different size (2188, 7149, 12.491, 17.225 and 22.021 m^2). The parameters of the warehouses are presented in Table 5.

Table 5. Differences between analyzed variants.

	Variant I	Variant II	Variant III	Variant IV	Variant V
Width (m)	53.0	100.7	119.3	137.8	159.0
Length (m)	41.3	71.0	104.8	125.0	138.5
Area of storage (m^2)	2188	7149	12.491	17.225	22.021
Maximum stock keeping unit	3410	10.035	18.835	23.435	31.035
Average stock keeping unit	2700	9376	17.696	21.589	28.276

For the variants presented in Table 5, products classification was done. On this basis, products allocation in the storage area was planned. The next stage of the research was the simulation of the products picking process following the model discussed above. Each simulation was performed for 1000 picking lists. In the example presented in this publication, 120 picking lists (6 lists per day for one month) had been used to teach the ANN. The picking lists were reflecting the most typical cases, and constituted representative data for the analyzed company. Subsequent picking lists were generated based on such data. To determine the size of the test sample it was checked whether the simulation results would be different depending on the number of picking lists used. Simulations were performed every 100 cases in the range from 100 to 3000. Statistical analysis evidenced that above 1000 picking lists, their increase did not affect the simulation result. Consequently, the size of the research sample for the model was set at 1000 picking lists for every simulation process.

The average time of order-picking referred to the model and classification criteria applied, as well as the warehouse size, as shown in Figure 9.

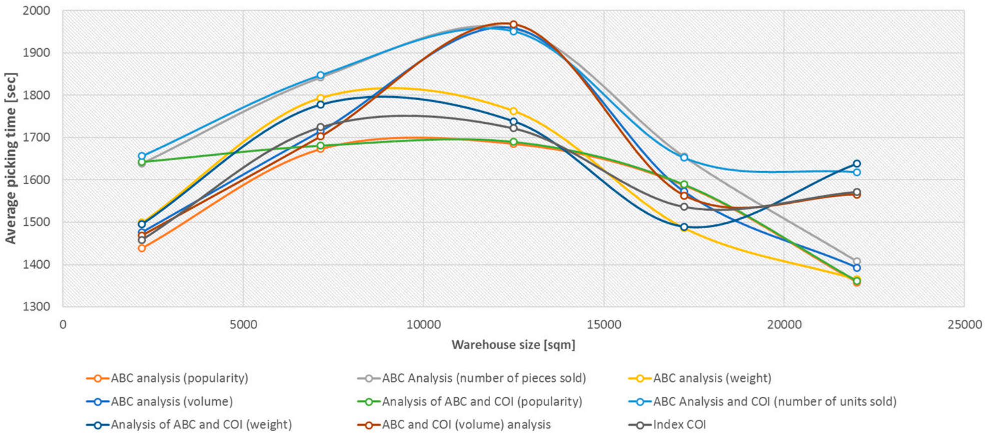


Figure 9. Average picking time depending on the warehouse size and classification method.

The simulations indicate that the picking time decreases for warehouse sizes larger than 10.000 m^2. The product picking times resulting from the simulations for two warehouse areas (2188 m^2, 7149 m^2) and nine (36) variants of product allocation were a basis for deriving linear functions, identifying the increase in order picking time depending on the warehouse area. They took the following forms:

$$\begin{aligned}
y &= 21.126x - 28{,}203\\
y &= 24.380x - 37{,}777\\
y &= 16.849x - 23{,}067\\
y &= 20.742x - 27{,}442\\
y &= 128.93x - 20{,}962\\
y &= 25.968x - 40{,}827\\
y &= 17.537x - 24{,}032\\
y &= 21.156x - 28{,}878\\
y &= 18.639x - 24{,}997
\end{aligned} \tag{36}$$

On the basis of the above functions the mean function was derived:

$$y = 32.810x - 49{,}428 \tag{37}$$

This function, in turn, was a basis for the forecast of picking time increase. The results of this forecast for each classification model and product allocation are tabulated in Table 6.

Table 6. Forecast of picking time depending on variants.

Classification Method	Picking Time [sec]				
	Variants				
	I	II	III	IV	V
ABC analysis (popularity)	1439	1673	1926	2150	2377
ABC Analysis (number of pieces sold)	1639	1843	2062	2256	2453
ABC analysis (weight)	1499	1793	2110	2391	2676
ABC analysis (volume)	1477	1668	1925	2153	2385
Analysis of ABC and COI (popularity)	1643	1681	1723	1760	1797
ABC Analysis and COI (number of units sold)	1656	1848	2053	2236	2420
Analysis of ABC and COI (weight)	1495	1778	2083	2353	2626
ABC and COI (volume) analysis	1468	1703	1955	2179	2406
Index COI	1459	1725	2011	2265	2523

The difference between the results (presented in percentage in Table 7 and Figure 10) and the forecast was calculated.

Table 7. Product picking time reduction vs. forecast (%).

Classification Method	Variants				
	I [+]	II	III	IV	V
ABC analysis (popularity)	0	3.05	11.95	27.98	60.39
ABC Analysis (number of pieces sold)	0	−6.42 *	−3.62 *	22.75	54.70
ABC analysis (weight)	0	−3.85 *	7.07	36.69	59.52
ABC analysis (volume)	0	0.49	−3.66 *	29.21	56.33
Analysis of ABC and COI (popularity)	0	2.56	11.65	27.83	60.03
ABC Analysis and COI (number of units sold)	0	−6.66 *	−3.27 *	22.92	34.55
Analysis of ABC and COI (weight)	0	−3.02 *	8.55	36.43	32.92
ABC and COI (volume) analysis	0	1.26	−4.11 *	29.98	39.13
Index COI	0	−0.02 *	9.60	32.24	38.54

[+] reference variant; * values higher than forecast.

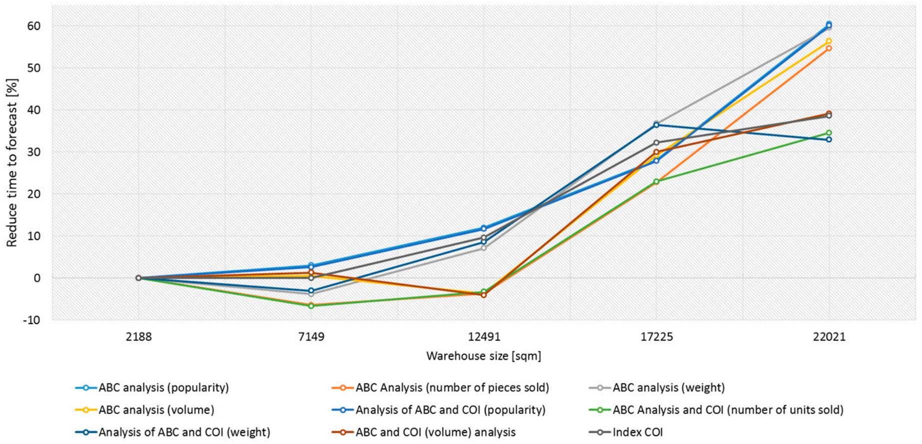


Figure 10. Products picking time reduction vs. forecast (%).

The most favorable results were obtained for the ABC analysis according to the popularity criterion. For this variant, detailed statistical analyses are presented, and the results are shown in Figure 11.

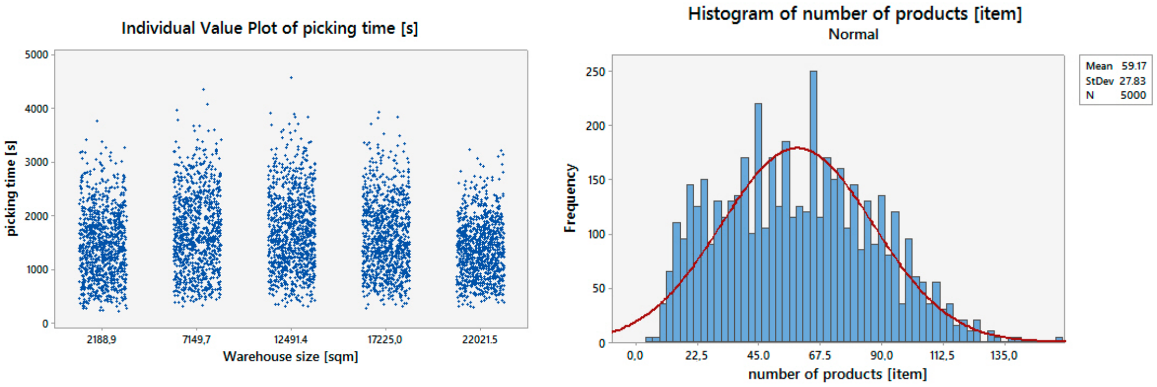


Figure 11. Order-picking time depending on warehouse size for ABC analysis after popularity criterion (**left**) and histogram of the number of products picked (**right**).

For all the methods, statistical tests were done to determine whether the results for particular warehouse sizes differ from one another. The most favorable results were obtained for the ABC classification according to product popularity. The results were further analyzed statistically. Table 8 shows various statistics for each of the five columns of data.

Table 8. Summary statistics for picking time (sec).

Area of Storage (m^2)	Average Picking Time (sec)	Standard Deviation	Coeff. of Variation	Minimum	Maximum	Range
2188	1438	643.6	44.74%	198.0	3753.0	3555.0
7149	1673	718.5	42.93%	303.0	4342.0	4039.0
12.491	1685	716.8	42.52%	290.0	4549.0	4259.0
17.225	1587	665.1	41.90%	265.0	3905.0	3640.0
22.021	1357	529.8	39.02%	272.0	3224.0	2952.0
Total	1548	670.8	43.32%	198.0	4549.0	4351.0

The ANOVA (Table 9) analysis divides the difference between the data into two parts: the between-group part and the within-group part. The F-ratio, which in this case equals 48.68, is the ratio of the between-group estimate to the within-group estimate. Since the *p*-value of the F-test is less than 0.05, there is a statistically significant difference between the means of the five variables at the 5% significance level.

Table 9. Results of ANOVA analysis.

Source	Sum of Squares	Mean Square	F-Ratio	*p*-Value
Between groups	8.44107×10^7	2.11027×10^7	48.68	0.0000
Within groups	2.16545×10^9	433,523		
Total (Corr.)	2.24986×10^9			

A multiple range test by Tukey HSD for the 95% confidence interval was done. Table 10 applies a multiple comparison procedure to determine which means are significantly different from which others.

Table 10. Multiple range tests for picking time (sec) by area of storage (m^2).

Variant	Area of Storage (sqm)	Count	Mean	Homogeneous Groups			
I	2188	1000	1438.5		x		
II	7149	1000	1673.3				x
III	12.491	1000	1685.7				x
IV	17.225	1000	1587.2			x	
V	22.021	1000	1357.7	x			

The bottom half of the output shows the estimated difference between each pair of means. Four homogenous groups are identified using columns of Xs. Within each column, the levels containing Xs form a group of means within which there are no statistically significant differences. The model currently being used to discriminate among the means is Tukey's honestly significant difference (HSD) procedure. With this method, there is a 5% risk of calling one or more pairs significantly different when their actual difference equals 0.

Based on the analyses performed, it was found that there is a statistical dependence of the picking time on the warehouse area. The results of the research presented can provide a solid basis to assist in choosing a product classification method for planning the distribution of products in a warehouse.

Based on the simulations for a small warehouse, i.e., allowing the storage of a maximum of 2700 pallet units, it was found that the largest difference between the median values occurs for the COI Index compared to the results obtained from the ABC analysis (2.56%) and the classical analysis (3.05). Therefore, it was found that for warehouses of the considered size, the selection of the product classification criterion is irrelevant—just a few per cent. The organization of the product storage system is also irrelevant—the allocation of storage areas or the free storage space method improves efficiency by only 2.5%. Therefore, for warehouses of a similar size, it is not possible to significantly improve the efficiency of its operation as a result of the application of product classification. In this case, it is more rational to reduce the costs of warehouse operation by reducing the storage area.

On the other hand, when conducting simulations for medium and large warehouses (Variant IV and V), it was found that the best results were obtained when the products were distributed in the warehouse based on ABC analysis (weight) and the analysis of ABC and COI (weight).

The worst results were obtained using the ABC analysis combined with the COI Index according to the popularity of products—the median value of the picking time was 1797 s, and the range of results was 8610 s. It can therefore be concluded that large warehouses and products with low susceptibility to damming loads are best suited to methods where the decisive criteria are the volumes and weights of products, such as the COI Index.

On the other hand, the use of the presented method allows the warehouse to operate more efficiently, by approximately 16.8%. This is a result more than twice as good as that which can be achieved with the use of classical classification methods, regardless of the adopted criterion. Due to the lower labor consumption and relatively high speed of computing, this method can be successfully used in warehouses, where the quantitative share of individual products and the assortment often change. Both the method of searching for links and the use of artificial neural networks can be successfully implemented for practical applications.

The developed method has a comprehensive approach, i.e., it allows taking into account the relationships between products, based on the orders. The analyses have proven that there is a statistical dependence of picking time on warehouse area size (Figure 12). The simulations indicate that the picking time decreases for warehouse sizes larger than 10.000 m^2. For small warehouses, the product choosing method for allocation plays a secondary role.

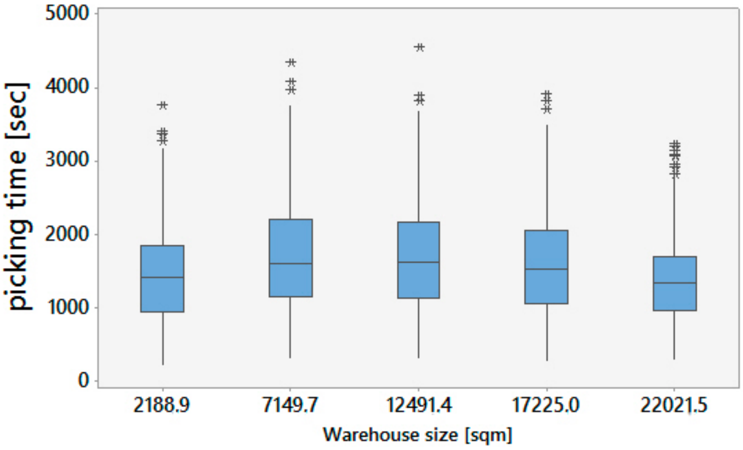


Figure 12. Picking time for different warehouse size (sec).

6. Conclusions

In the proposed research, a new model was developed and implemented as a software called Pickup Simulo, which was presented along with a discrete event simulation analysis. The model is quite complex, and not only does it contain methods of ANN, but it also enables the classification of products by ABC analysis according to the selected criteria, the definition of COI and, finally, product allocation planning. Based on the results of our case study, a decrease from 10 to 16% of the total order-picking time can be expected by using our model. Implementation of the presented model enables us to find the optimal location for products, so the distance covered in product picking is reduced. This results in the increased efficiency of order-picking.

The research results are, therefore, of significant importance for warehouse managers. They can thus be used in companies which use standard warehouse systems with storage racks. Moreover, the discussed model could be used for the evaluation of potential solutions before their practical implementation in a warehouse, which will, consequently, result in reducing the risk of implementing an ineffective solution. The cost of the order-picking process is approximated at 30 to 50% of all warehouse costs. Because of this, product layout planning plays an important role in operational decisions. The model can support that decision, and reduce the risk of a bad decision in product location planning. At present, the proposed model Pickup Simulo is being used in selected companies. Based on the results of the proposed research, the authors' model and its implementation in the software called Pickup Simulo could be helpful in the selection of a product classification model when planning product allocation in a new warehouse.

In future works, the authors want to upgrade the proposed model of Pickup Simulo with the possibility of simulating different warehouse systems.

Author Contributions: Conceptualization, A.L.; methodology, A.L.; software, A.L.; validation, A.L.; formal analysis, A.L.; investigation, A.L.; resources, A.L.; data curation, A.L. and T.L.; writing—original draft preparation, A.L. and T.L.; writing—review and editing, A.L. and T.L.; visualization, A.L.; supervision, T.L.; project administration, A.L.; funding acquisition, A.L. and T.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: This work was supported by project: M-8/355/2016/DS-M: Optimization of logistics processes in warehouse.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

w_t	the average number of product units sold
l_p	the average number of employees in the warehouse, storekeeper
d_{cr}	the aisle width
D_{lr}	the rack length
dr_{poprz}	the length of a route across the warehouse
d_{rwzd}	the length of the route along the warehouse
D_{wr}	the rack width
i	the product index
I_c	the correction indicator
id_{prody}	the product ID (corresponding to the sum s_{uy})
idx	matrix A column index
idy	matrix A row index
I_p	the intensity work of employees, loading unit/man-hour or kg/man-hour
k	correlation coefficient of the results for input parameter analysis
kat_{ABC}	category obtained by application of the ABC analysis by weight
kat_{ABC2}	category obtained by application of the ABC analysis by volume
kat_{AP}	category obtained by application of the algorithm of relationships search
kat_{COI}	category obtained by application of the COI Index
K_p	the cost of pallets flow through the warehouse, EUR/quantity
K_{rm}	the annual cost of storage, EUR
l_h	the number of hours worked by storekeeper, man-hour
l_{korwz}	total value of longitudinal aisle, rounded up
l_p	the number of storekeepers, workers
l_{reg}	the number of racks in a row
l_{sp}	the number of items picked, quantity
l_{sp}	the number of picked lines in order, quantity
l_{spp}	the number of items correctly picked, quantity
max_range	maximum range of values for row to be removed (default is 0)
n	number of analyses/periods
p_i	demand: product popularity or average demand, units
P_k	the correctness of picking
P_{pmax}	the maximal value

P_{pmin}	the minimal value
P_q	the product quantity
P_r	an annual flow of units through the warehouse, quantity
PS	process settings that allow consistent processing of values
s_i	standard deviation
t_a	an additional time of penalty for forklift turn
t_{lp}	the time of moving 1 m straight route
t_p	the time of lifting and lowering fork on specified stock level (p)
t_{pp}	total moving time
t_{pr}	work time per shift, h
V_i	coefficient of variation
v_i	product size: volume or weight, m^3, kg
w	weighting coefficient
w_i	COI Index
w_{ij}	number of *i*-product units sold in the *j*-period
W_k	the picking productivity, quantity/man-hour
w_{om}	the turnover of a warehouse (as the sum of delivery and dispatch), quantity or kg
W_p	the work productivity, loading unit/storekeeper or kg/storekeeper
X	N-by-Q matrix, X = *removeconstantrows('reverse',Y,PS)* returns X, given Y and settings PS
x_{konc}	coordinate of destination rack in a row
x_{pocz}	coordinate of an actual rack in a row
x_{temp}	route length between coordinates for racks in a row
Y	M-by-Q matrix with N–M rows deleted, Y = *removeconstantrows('apply',X,PS)* returns Y, given X and settings PS
y_{konc}	coordinate of destination row
y_{pocz}	coordinate of actual row
y_{temp}	route length between coordinates for row

References

1. Li, M.-L. Goods classification based on distribution center environmental factors. *Int. J. Prod. Econ.* **2009**, *119*, 240–246. [CrossRef]
2. Clott, C.; Hartman, B.C. Supply chain integration, landside operations and port accessibility in metropolitan Chicago. *J. Transp. Geogr.* **2016**, *51*, 130–139. [CrossRef]
3. Subramanian, N.; Ramanathan, R. A review of applications of Analytic Hierarchy Process in operations management. *Int. J. Prod. Econ.* **2012**, *138*, 215–241. [CrossRef]
4. Henn, S. Algorithms for On-line Order Batching in an Order-picking Warehouse. *Comput. Oper. Res.* **2012**, *39*, 2549–2563. [CrossRef]
5. Chen, F.-L.; Chen, Y.-C.; Kuo, J.-Y. Applying moving back-propagation neural network and moving fuzzy neuron network to predict the requirement of critical spare parts. *Expert Syst. Appl.* **2010**, *37*, 4358–4367. [CrossRef]
6. Rao, S.S.; Adil, G.K. Class-based storage with exact S-shaped traversal routeing in low-level picker-to-part systems. *Int. J. Prod. Res.* **2013**, *51*, 4979–4996. [CrossRef]
7. Hadi-Vencheh, A.; Mohamadghasemi, A. A fuzzy AHP-DEA approach for multiple criteria ABC inventory classification. *Expert Syst. Appl.* **2011**, *38*, 3346–3352. [CrossRef]
8. Moeller, K. Increasing warehouse order-picking performance by sequence optimization. *Procedia Soc. Behav. Sci.* **2011**, *20*, 177–185. [CrossRef]
9. Chu, C.-W.; Liang, G.-S.; Liao, C.-T. Controlling inventory by combining ABC analysis and fuzzy classification. *Comput. Ind. Eng.* **2008**, *55*, 841–851. [CrossRef]
10. Altay Guvenir, H.; Erel, E. Multicriteria inventory classification using a genetic algorithm. *Eur. J. Oper. Res.* **1998**, *105*, 29–37. [CrossRef]
11. Van Den Berg, J.P.; Zijm, W.H.M. Models for warehouse management: Classification and examples. *Int. J. Prod. Econ.* **1999**, 59. [CrossRef]
12. Yu, M.C. Multi-criteria ABC analysis using artificial-intelligence-based classification techniques. *Expert Syst. Appl.* **2011**, *38*, 3416–3421. [CrossRef]

13. Muppani, V.R.; Adil, G.K. Efficient formation of storage classes for warehouse storage location assignment: A simulated annealing approach. *Omega* **2008**, *36*, 609–618. [CrossRef]
14. Boysen, N.; Stephan, K. The deterministic product location problem under a pick-by-order policy. *Discret. Appl. Math.* **2013**, *161*, 2862–2875. [CrossRef]
15. Roodbergen, K.J.; Vis, I.F.A.; Taylor, G.D. Simultaneous determination of warehouse layout and control policies. *Int. J. Prod. Res.* **2015**, *53*, 3306–3326. [CrossRef]
16. van Geest, M.; Tekinerdogan, B.; Catal, C. Design of a reference architecture for developing smart warehouses in industry 4.0. *Comput. Ind.* **2020**, *124*, 103343. [CrossRef]
17. Bandaly, D.; Satir, A.; Shanker, L. Impact of lead time variability in supply chain risk management. *Int. J. Prod. Econ.* **2016**, *180*, 88–100. [CrossRef]
18. Mohammed, A.; Harris, I.; Soroka, A.; Naim, M.; Ramjaun, T.; Morteza, Y. Gresilient supplier assessment and order allocation planning. *Ann. Oper. Res.* **2020**, 1–28. [CrossRef]
19. Ballesteros-Riveros, F.A.; Arango-Serna, M.D.; Adarme-Jaimes, W.; Zapata-Cortes, J.A. Storage allocation optimization model in a Colombian company. *DYNA* **2019**, *209*, 255–260. [CrossRef]
20. Li, X.; Zheng, Y.; Zhou, Z.; Zheng, Z. Demand Prediction, Predictive Shipping and Product Allocation for Large-Scale E-Commerce. 2019. Available online: https://ssrn.com/abstract=3277125 (accessed on 12 March 2019).
21. Zapata-Cortes, J.A.; Arango-Serna, M.D.; Serna-Urán, C.A.; Adarme-Jaimes, W. Mathematical Model for Product Allocation in Warehouses. In *Techniques, Tools and Methodologies Applied to Global Supply Chain Ecosystems*; García-Alcaraz, J., Sánchez-Ramírez, C., Avelar-Sosa, L., Alor-Hernández, G., Eds.; Springer: Heidelberg, Germany, 2020; Volume 166.
22. Guerriero, F.; Musmanno, R.; Pisacane, O.; Rende, F. A mathematical model for the Multi-Levels Product Allocation Problem in a warehouse with compatibility constraints. *Appl. Math. Model.* **2013**, *37*, 4385–4398. [CrossRef]
23. Guerriero, F.; Pisacane, O.; Rende, F. Comparing heuristics for the product allocation problem in multi-level warehouses under compatibility constraints. *Appl. Math. Model.* **2015**, *39*, 7375–7389. [CrossRef]
24. Soleimaninanadegany, A.; Hassan, G.A.; Rahiminezhad, M. Product allocation of warehousing and cross docking: A genetic algorithm approach. *Int. J. Serv. Oper. Manag.* **2017**, *27*, 239–261.
25. Zhou, G.; Min, H.; Gen, M. A genetic algorithm approach to the bi-criteria allocation of customers to warehouses. *Int. J. Prod. Econ.* **2003**, *86*, 35–45. [CrossRef]
26. Zhu, S.; Hu, X.; Huang, K.; Yuan, Y. Optimization of product category allocation in multiple warehouses to minimize splitting of online supermarket customer orders. *Eur. J. Oper. Res.* **2020**. [CrossRef]
27. Sremac, S.; Tanackov, I.; Kopić, M.; Radović, D. ANFIS model for determining the economic order quantity. *Decis. Mak. Appl. Manag. Eng.* **2018**, *1*, 81–92. [CrossRef]
28. Lorenc, A.; Kuźnar, M. An Intelligent System to Predict Risk and Costs of Cargo Thefts in Road Transport. *Int. J. Eng. Technol. Innov.* **2018**, *8*, 284–293.
29. Barreto, S.; Ferreira, C.; Paixão, J.; Santos, B.S. Using clustering analysis in a capacitated location-routing problem. *Eur. J. Oper. Res.* **2007**, *179*, 968–977. [CrossRef]
30. Chan, F.T.S.; Chan, H.K. Improving the productivity of order-picking of a manual-pick and multi-level rack distribution warehouse through the implementation of class-based storage. *Expert Syst. Appl.* **2011**, *38*, 2686–2700. [CrossRef]
31. Lorenc, A.; Lerher, T. Effectiveness of product storage policy according to classification criteria and warehouse size. *FME Trans.* **2019**, *47*, 142–150. [CrossRef]
32. Straka, M.; Lenort, K.; Feliks, J. Design of large-scale logistics systems using computer simulation hierarchic structure. *Int. J. Simul. Model.* **2018**, *17*, 105–118. [CrossRef]
33. Caron, F.; Marchet, G.; Perego, A. Routing policies and COI-based storage policies in picker-to-part systems. *Int. J. Prod. Res.* **1998**, *36*, 713–732. [CrossRef]

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.