drones

MDPI

Article

A Path Planning Model with a Genetic Algorithm for Stock Inventory Using a Swarm of Drones

Miklós Gubán and József Udvaros *

Faculty of Finance and Accountancy, Budapest Business School, 1149 Budapest, Hungary
* Correspondence: udvaros.jozsef@uni-bge.hu

Abstract: In this paper, a mathematical model and solution for performing the inventory tasks of a multi-user, mixed warehouse in which neither satellite positioning nor other IT solutions can be used was presented. After reviewing the literature on road planning and the use of drones in warehouses, a method is presented that can be used to control drones that can be moved in all directions for imaging and transmission. The proposed method consists of three main steps. As a first step, we provide the mathematical model and solution method needed to determine the (optimal execution time) access routes required for processing the compartments of the warehouses. This is an initial step before starting the inventory. This considers the structure of the warehouse, its features, the number of drones, and the parameters of the drones. In the second step, based on the routes obtained in the first step, the real-time movement of the drones was controlled during processing, including camera movement and image recording. The third step is post-processing, i.e., processing the images for QR code identification, interpreting the QR code, and recognizing empty compartments for inventory control. A major advantage for users of the solution method is that the result can be achieved automatically without an external orientation device, relying solely on its own movement and the organization of a pre-planned route. The proposed model and solution method are suitable not only for inventory control, but also for solving other problems matching the model.

Keywords: stock inventory; drone; genetic algorithm

check for updates

Citation: Gubán, M.; Udvaros, J. A Path Planning Model with a Genetic Algorithm for Stock Inventory Using a Swarm of Drones. *Drones* **2022**, *6*, 364. https://doi.org/10.3390/drones6110364

Academic Editor: Xiwang Dong

Received: 25 October 2022
Accepted: 17 November 2022
Published: 19 November 2022

1. Introduction

The traceability of a company's products and inventory is now a fundamental factor in a company's daily operation. In this regard, inventory management is extremely important, as each item in inventory represents part of the value of the business, and it is important for managers to know the value of the items in the corporate organization they lead. Thus, from a strategic point of view, it is very important to introduce the correct practices of stock management, so that the logistics managers of the company know and are able to manage the material flow processes of the organization. Certain tasks of stock management require high levels of human resources (inventory, order picking, etc.) [1]; therefore the training of specialists cannot be neglected, with a special focus on the development of computer thinking [2].

Drones are a breakthrough technology in warehouse inventory management as they enable real-time, fast, and accurate inventory records while minimizing operational costs for supply chain stakeholders. From a sustainability perspective, drones socially minimize the risk of human injury in the facility. From an environmental perspective they use environmentally friendly devices, i.e., rechargeable batteries, to navigate the facility layout and execute the daily operational schedule. From an economic perspective, to enable near-continuous inventory control and optimize operational decision-making, they minimize labor costs and operational downtime related to inventory [3]. Drone-enabled supply chain solutions are based on first mile and last mile delivery. Their efficiency significantly affects the total operating cost. Drone technologies also enable the improvement of first mile delivery and last mile operations, but the design and optimization of these solutions mean new challenges [4]. Drone-based material handling is becoming increasingly important today, as advances in technology and increasingly sophisticated regulation allow drone transport to be used in more and more areas of practice. The deployment of drone systems requires a number of planning tasks that require rethinking logistics tasks such as route planning, scheduling, and resource allocation [5,6].

Drones can help the industry—in addition to the previous ones—perform tasks that can be automated and monotonous, such as maintaining product traceability. Therefore, the system uses a distributed ledger to store, to validate, and ensure the reliability of inventory data collected by drones and make it available to interested parties [7]. In order to demonstrate the performance of the system proposed by the authors of [8], various tests were carried out in a real industrial warehouse, with the following conclusion: the system is able to obtain inventory data very quickly compared with traditional manual tasks, while it is able to estimate the position of the items thanks to the signal strength of the tags when the drone hovers over the elements.

In this article, we presented a mathematical model and procedure not encountered in our literature review. The model and method describe the inventory of a GPS-inaccessible high warehouse with multiple drones using QR code for a multi-user warehouse. The complexity of the problem is discussed later in this paper.

2. Literature Review

Inventory work is an important link in the invoicing system of industrial enterprises. Considering the current shortcomings of manual inventory work, such as low efficiency and potential for error, technology has emerged in recent years that has allowed drones to be equipped with high-precision portable radio-frequency identification readers for inventory, and the key issue is route planning. In order to reduce energy consumption and improve inventory efficiency, the lowest energy efficiency ratio and time efficiency ratio were taken as the objective to construct the objective function, and the corresponding mathematical model of UAV route planning is established in literature [9]. In the model given by ref. [9], a hybrid differential evolution (DE) algorithm based on the life-cycle swarm optimization (LSO) algorithm was proposed. The actual environmental data of a tobacco company's raw material and auxiliary warehouse were used for the physical modeling, and the simulation experiment was performed using the proposed intelligent optimization algorithm. In the same environment, compared with the other three leading path planning algorithms, the experimental results showed that the path planned by the proposed algorithm can effectively improve the efficiency of UAV storage.

During the inventory, an important issue is raised: the access route. Nowadays, many articles deal with drone routing, which is closely related to solving the inventory problem. During our literature research, different methods were used for drone route planning. Many articles deal with route planning based on a two-dimensional environment, which is not suitable for practical application, since our shelving system requires spatial route planning. These two-dimensional solutions provide a suitable solution mostly for agricultural problems and patrol and monitoring tasks. The following algorithms are most often used by researchers during route planning suitable for inventory: the traditional squid algorithm [10]; the genetic algorithm [11,12]; the rapidly expanding random tree algorithm [13,14]; the traditional particle swarm optimization algorithm (PSO); the ant colony algorithm [15,16]; and deep neural networks [17,18]. Most of the research only uses the shortest flight path [19] or the least energy consumption [20] to generate the single-objective optimization model, and further uses the convex approximation strategy [21] to calculate the route. These solutions can be used from a practical point of view, as examined in this study. However, our target function was a max–min time function, which does not allow the direct application of the specified methods.

The key to drone routing and effective inventory is for the drone to carry a high-precision, portable, radio-frequency identification (RFID) reader to perform an inventory task according to ref. [22]. Based on a quadrotor drone, they proposed a route-planning method that uses a drone equipped with an RFID reader to carry out inventory of industrial product warehouses. Since the particle swarm optimization algorithm (PSO) tends to converge prematurely when solving routing problems and tends to fall into local optima (which, according to our experience [23], often happens), PSO is improved and a differential evolution based a repair method is suggested. The fitness-adaptive differential evolution algorithm (FiADE) and PSO were mixed (used previously [24,25]) and further developed for further application in three-dimensional space. The final simulation results showed that the improved PSO-based hybrid suitability DE algorithm (PSO-DE) has higher uniformity than the DE, PSO, and whale optimization algorithms (WOA), and is more suitable for drone route planning in complex industrial warehouses.

In order to solve the inventory task, it is necessary to examine which drones are suitable for performing the task. Our previous investigations [23] showed that autonomous drones are needed for the task. For a drone to be fully autonomous, the ability to localize is very essential. Adequate accuracy is usually achieved by using several sources of positional information. During the research [26], two sources of information were used:

- The Intel RealSense T265 tracking camera to calculate continuous visual odometry, and
- The RGB sensor combined with the ArUco reference indicators to minimize the accumulated error during visual odometry.

These markers have their mistakes. A variance filter was successfully implemented and evaluated to minimize the error caused by ArUco markers. The entire system was tested several times on a drone that can fly autonomously in an indoor environment. Furthermore, all processes required for such localization ran smoothly on the Nvidia Jetson Nano computer mounted directly on the drone during flight. All these results proved to be a very strong foundation for an autonomous drone that can perform warehouse inventory in an indoor environment, all without human intervention.

According to the literature [27], the goal of introducing an unmanned aerial vehicle stem UAV (drone) in a warehouse is to reduce operation times and eliminate human labor in repetitive, simple tasks. One such area is the inventory process, which are several information carriers (QR codes (Quick Response), barcodes, or RFID tags) on the goods in the warehouse. This process is particularly labor-intensive in large, high-shelf warehouses where thousands of products are stored. Not every business needs a daily inventory requirement. Most warehouses perform this process periodically (monthly, quarterly, etc.). However, such an inventory is always a challenge, as the warehouse operations must be suspended and the entire warehouse staff must be involved in the inventory process, which is a loss for the company. Because of this, there is a growing need to automate this process to reduce time and reduce human error.

In ref. [28], multi-sensor accuracy was investigated for 3D indoor positional accuracy using the OASE flexible sensor fusion platform. This evaluation was based on real drone flights in an industrial laboratory using mm-accurate measurements provided by motion capture cameras, allowing sensors to be evaluated based on their deviation in 2D and 3D. The research investigated the following sensors: IMU, sonar, SLAM camera, ArUco markers, and Ultra-Wideband (UWB) positioning with up to six anchors. The paper demonstrated that the achievable 2D (3D) indoor localization error varied from 4.4 cm to 21 cm (4.9 cm to 67.2 cm) depending on the sensor set selected. Furthermore, cost/accuracy trade-offs were defined to indicate the relative importance of different sensor combinations depending on the (technical) budget and use case. These laboratory results were validated during a “Proof of Concept” installation of an inventory drone in a 65,000 m^2 warehouse with more than 10 flight hours. By combining laboratory results and real-world deployment experience, different subsets of sensors were presented as minimally feasible solutions for three different indoor use cases, considering accuracy and cost: a large drone with low weight and cost constraints, one or more medium-sized drones, and a weight, and a swarm of cost-constrained nanodrones. This is a valuable study. The GPS positioning system cannot be used in the warehouse we are examining, so the routes must be determined in advance. Based on the results of the study, we have to make sure that if the drone flies in the middle of the aisle, i.e., it cannot collide with the shelf system due to the inaccuracy of the sensors. Of course, a certain flight distance requires the drone to be calibrated and directed to a given optimal position.

One of the biggest challenges for indoor drone applications is navigation and obstacle avoidance. Due to indoor operations, the global positioning system is not capable of accurate localization and navigation. To solve this problem, ref. [29] presented a system that facilitates the autonomous navigation of drones with cameras in the interior aisles of a building by processing images based on deep neural networks. For a deep neural network, choosing a good combination of hyper-parameters for better prediction is a complicated task. In the mentioned article, the tuning of the hyper-parameters of a convolutional neural network was implemented using genetic algorithms. The proposed architecture (DCNN-GA) was compared with state-of-the-art ImageNet models. The experimental results showed minimal loss and high performance of the proposed algorithm. Similar results were obtained by De Falco in ref. [30]. In their article, they proposed an approach based on the use of unmanned aerial vehicles for warehouse scanning and the use of region-based convolutional neural network (R-CNN) for autonomous inventory activities. The experimental results obtained from the video recordings of the real warehouse environment proved the feasibility of the proposed solution and the possible scope for development.

There are also other approaches for path planning and optimization, such as that presented Nguyen et al. in ref. [31], who combined fuzzy logic and ANFIS approaches for optimization. The optimization of Wang et al. followed an artificial intelligence algorithmic approach [32].

To automate inventory management in a large warehouse, a drone-mounted camera scans a pre-displayed ground QR code to reveal the route. The drone runs along the navigated route and manages the inventory of the warehouse by scanning the barcode or QR code attached to the product. However, unlike warehouses, which have well-defined grids or shelves, the location of products in a yard is not fixed, but flexible. Thus, for effective inventory management in a warehouse, the position of the QR codes attached to the product must also be estimated. Yoons [33] proposed a position estimation method for drones and products based on the segmentation model of QR codes in their research. The segmentation model was used to recognize the region of the QR code with perspective distortion caused by the angle difference between the camera and the QR code. After that, the detected QR code region was rectified and decoded to determine whether it is a terrestrial QR code, and the position of the drone was estimated. Finally, the 3D coordinates of the QR code attached to the product, rather than the QR code on the ground, were calculated from the images taken by the drones from two different viewpoints. Consequently, the 3D position coordinates of the drones and the QR codes attached to the products were estimated using the QR codes on the ground, and thus effective inventory management was realized in the warehouse. In our case, we worked with a complex warehouse structure, unlike ref. [23]. The structure of the warehouse can be recognized with a preliminary structure recognition algorithm using the previous ones. This is, however, beyond the scope of this article.

Kessler, in ref. [34], presented an approach for the automation of stock inventory using a drone equipped with a camera and various artificial intelligence procedures. This avoids the use of sensor technology such as RFID, and only visual display of products and goods was used. A unique data set was developed and used to train an object-recognition model to extract and count all relevant objects based on an image of the warehouse. It was shown that different pre-processing steps and especially image enhancement methods can significantly affect the performance of such models.

Several specialized studies deal with solutions that use RFID technology during the inventory process [1,8,35–37]. Others use barcodes to automatically identify data as a solution for drone inventory [38,39].

A model that uses an omnidirectional drone capable of moving in all directions for imaging and transmission was presented in ref. [23]. The proposed model includes three steps. In the first step, an optimal solution for determining a transit route was provided. This is consistent with the structure and capabilities of the warehouse. In the second step, the real-time movement of the drone was determined during processing, including camera movements and image capture. The third step is post-processing, i.e., processing the images for QR code identification, QR code interpretation, and checking matches and discrepancies for inventory control. A key advantage is that the result can be achieved without external orientation devices, relying solely on its own movement and the organization of the pre-planned route. The proposed model can be effective not only in inventory control, but also in revealing the structure of a warehouse shelf system and determining empty compartments. The article presented the model and solution method for traversing shelf systems with fixed positions using a drone. The solution we outlined is a significant further development of the model presented in ref. [23]. At the end of the Artificial Intelligence (GA) solution used in ref. [23], it is advisable to carry out tests such as those in ref. [40].

Shape recognition is a useful tool for many engineering tasks, such as structural inspection, traffic monitoring, or even QR code recognition. A new shape recognition technique was proposed for the recognition of shapes based on their contours, which can be used for fast processing of drone images. The image was first segmented into regions, then the contours were analyzed, and shape descriptors were built, which were used as inputs to the classifier. Simplified sequential fuzzy indexing tables were used for pattern recognition, so that very fast and robust inferences could be made [41–43].

3. Materials and Methods

In the large, lightweight warehouses of logistics centers, especially those where different products of several companies are stored, it is often difficult to determine the exact location of the stored goods. This mainly occurs if the storage is not carried out with an automatic forklift or when, due to a mistake, an error occurs in the registered and actual location of the goods during picking and put away. To make matters worse, it is very difficult or in some cases impossible to use GPS-based identification in these warehouses. If time-of-flight (ToF) cameras are not available, automatic device positioning is not possible. In this study, a model was presented that can update the stock register with a drone, and that can be generally applied to different warehouse structures.

Storage (loading and unloading), order picking, unit load design, and labeling take place in the warehouse building. A double shelving system is used in the warehouse. Between the double shelves, there is a wide aisle where the drone can travel. A shelf system is divided into rows or compartments. One, two, or three spaces can be created in one compartment (see Figure 1). All units stored in these locations were identified with a QR code that contains all the important data for their management.

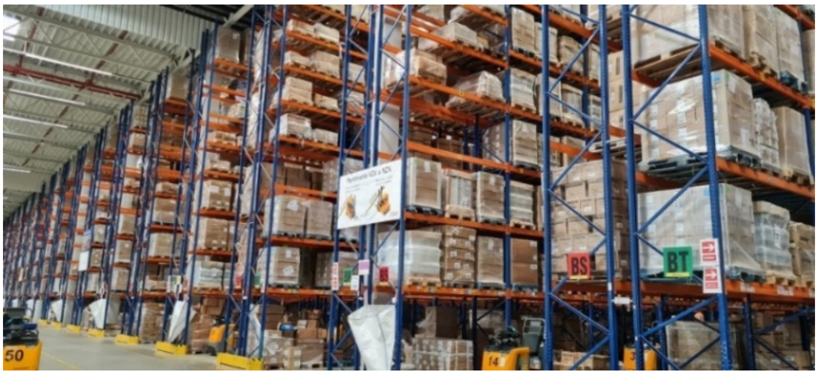

Figure 1. Construction of a specific warehouse.

A mid-range drone and its camera can take a high-quality image from the middle of the aisle (in the model to be described later, the drone was always set in the middle of the compartment), and the blue and red colors used on the shelves help to identify the stored goods in the image. At the end of the aisle, the docking station was placed manually, where the battery could be charged and replaced. However, its position must be specified exactly for each drone (see Table 1 parameter $\mathbf{D}_b = \mathbf{D}_b\left[d_{bx}, d_{by}, d_{bz}\right]$).

Table 1. Basic data of the warehouse and the drone.

Data (Warehouse)	Description
$\mathbf{w} = \mathbf{w}\left[w_x, w_y\right]$	the size of the warehouse (in meters).
$R_k = R_k(i, j, \beta)$	This indicates the compartment located in the *i*-th row and the *j*-th column of the two opposite shelf systems of the *k*-th aisle $(k = 1; 2; \ldots; r)$. Among the opposite shelving systems, the numbering starts with the first column of the left-hand shelving system in the case of columns, and then continues the numbering from the front (from 1) with the first column of the right-hand shelving system. $\beta = 0$, if left-hand shelf system and $\beta = 1$ if right-hand shelf system. This is a determining parameter due to the rotation of the camera.
$h_e(k, i, j, \beta)$	Function specifying the vertical distance between the compartments (calculated from given data).
h_h	The width of the compartment (in meters).
h_d	The depth of the compartment (in meters).
h_w	the width of the aisle (in meters).
h_c	The width of the road (charging aisle) (in meters).
h_g	The required height difference of the drones when the drone is on the changing aisle (in meters).
h_b	The height of the *b*-th drone when changing aisle (aisle change height, in meters). $h_b = b \cdot h_g$
$m_{kj\beta}$	The number of the compartments within a column (within the *j*-th column of the *k*-th aisle, β side). This and the height of the compartments ensure that shelf systems of different heights can be managed together.
$\hat{m}$	The number of all compartments ($\hat{m} = \sum_{k=1}^{r} \sum_{\beta=0}^{1} \sum_{i=1}^{n} \sum_{j=1}^{m_{ki\beta}} 1$).
n	The number of compartments within the rows. They can also be split, with different heights.
r	The number of aisles (the number of shelf systems is equal to twice the number of aisles).
$\mathbf{P}_l = \mathbf{P}_l\left[p_{lx}, p_{ly}, 0\right]$	$\mathbf{P}_l$ The shelf system's position. If l is odd, it means the left shelf system of the aisle $(\beta = 0)$; if it is even, it means the right shelf system $(\beta = 1)$.
$\mathbf{C}_{kb} = \mathbf{C}_{kb}\left[c_{kbx}, c_{kby}, c_{kbz}\right]$	The starting position of the drones in the aisle between the two shelving systems (viewed from the side of the main entrance). $c_{kbx} = h_d + \frac{h_w}{2} + (k-1) \cdot (h_w + 2h_d)$, $c_{kby} = h_c$. $c_{kbz} = h_b$. Each drone will have a starting position at a different height to avoid collision. $\mathbf{C}_{kb}$ can be calculated from the basic data.
$\overline{\mathbf{C}}_{kb} = \overline{\mathbf{C}}_{kb}\left[\overline{c}_{kbx}, \overline{c}_{kby}, \overline{c}_{kbz}\right]$	The end position of the aisle between the two shelves. $\overline{c}_{kbx} = h_d + \frac{h_w}{2} + (k-1) \cdot (h_w + 2h_d), \overline{c}_{kby} = n \cdot h_h + h_c$. $\overline{c}_{kz} = h_b$. $\overline{\mathbf{C}}_{kb}$ can be calculated from the basic data.
Date (drone)	Description
g	the number of drones

Table 1. *Cont.*

Data (Warehouse)	Description
$\mathbf{D}_b = \mathbf{D}_b\left[d_{bx}, d_{by}, d_{bz}\right]$	The position of the docking station of the b-th drone's. $d_{bz} = 0$.
T_w	The operating time of the drone when fully charged. During this time, the drone must proceed from the charger to the compartments and back.
T_{PH}	The photo shooting time.
V_h	The horizontal travel speed. The maximum travel speed is 16 m/s, but in practice it can be set at 10 m/s.
V_a	The rate of rise.
V_d	The rate of descent.
V_r	The rotation speed (estimated).
Δt	This indicates the operating time that must be less than when arriving at the charger, in case of intermediate charging.
T_{ch}	The charging time.

In the warehouse, processing with several drones at the same time can be carried out. In this case, each drone would work independently. However, each aisle can only be processed by one drone, i.e., two different compartments of an aisle cannot be processed by two different drones in the same processing cycle. In addition, to avoid collisions, the drones travel between aisles at different heights. The operation process for one drone is described below, which is applicable for all drones. As they are working simultaneously, the pre-processing system must handle all the drones participating in the processing at the same time.

3.1. The Course of the Processing

Moving along the aisle, the drone looks for the compartments and takes a photo of each compartment, which is sent to a pre-processing device (in our study, this was a high-capacity tablet) where the pre-processing application runs. During the pre-processing, the first test checked the quality of the image (in terms of brightness, sharpness, distracting factors). The drone did not change its position until the pre-processing application instructed it to do so. If the picture quality was not good enough, the pre-processing application took another picture with the drone. If the picture was good quality, the drone continued its journey. During the image processing, a proper QR code was being checked. If the QR code was appropriate, its content was read out. If successful, it sent the content of the QR code to the central application and set the processing status to 'successful'. If it was not successful, it sets the bin's processing status to 'failed'. If a QR code was not found, an empty compartment comment was entered (of course, this does not mean that the compartment is empty in all cases; however, the central application decided whether it is empty or due to incorrect processing). In this case, it always sent the drone to the next compartment. Further processing always took place in the central application and depended solely on the capabilities of the central application. The description of the processes taking place here is beyond the scope of this study.

After processing each compartment, the pre-processing application sent the position of the compartment (see Table 1 $R_k = R_k(i, j, \beta)$ values), the processing status, and the data in the QR code to the central application.

According to the model, a pre-processing system was proposed that knows the location of the drone in the warehouse exactly, i.e., the 3D data inside the warehouse are defined. This and the QR code sent can tell the inventory management model exactly where the product is located or if the location is empty.

The pre-processing application also includes the previously determined optimal routes for each drone, which are primarily optimized for the drones' operating time, including the charging time in case the drones run out of power.

Based on the above, the processing consists of three main steps (Figure 2).

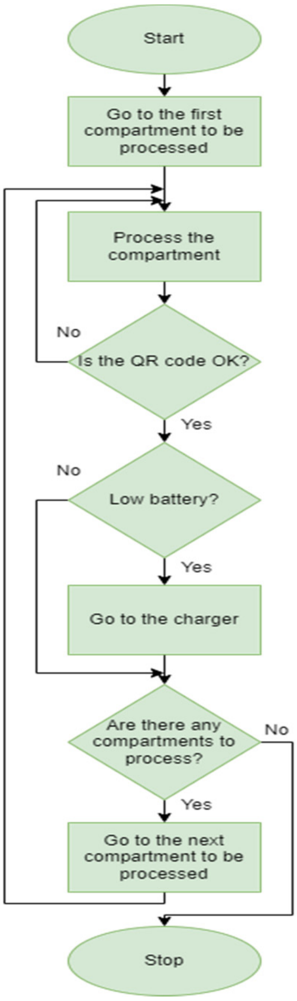
Start
Go to the first compartment to be processed
Process the compartment
No
Is the QR code OK?
Yes
No
Low battery?
Yes
Go to the charger
Are there any compartments to process?
No
Yes
Go to the next compartment to be processed
Stop

Figure 2. Flowchart of the processing on a drone.

Step 1: (initial step before running) determining the optimal route of the drones according to the shortest processing time. This is created by the central application and sent to the pre-processing application.

Step 2: (runtime processing in the pre-processing application) controlling the drones along the specified optimal paths to the compartments, taking the photos.

Step 3: (post-processing) to perform the pre-processing (photo processing) for the next step of the drone.

Based on the literature review, we did not come across an approach like ours for inventory registration; therefore, it can be declared that our proposed method for inventory can be considered a research-based new development in the ongoing debate and studies in this field.

Research questions

Our study was based on the following research questions:

- How can the stock inventory be managed with drone automation?
- Can the inventory process with drones be modeled mathematically?
- How can the stock inventory check be optimized with drones?

Regarding the first step, a mathematical model and algorithm was developed that accurately answers the abovementioned questions; in this way, a usable method for processing automated warehouse stocking was developed.

The solution method of Step 2, as the result of the first step is known, was already described in this chapter; it must be programmed in the pre-processing application based on what is described.

The first step in model creation is to clarify the markings and their content; therefore, full parameter exploration was performed. The dataset was defined with the help of practical experts. The basic data collected were grouped according to their role in the model. Then, after selecting the necessary parameters, it is important to fix the dimensions. In the second step, the mathematical model of the warehouse and the factory parameters of the drones to be used were prepared and supplemented with variable data. A manageable simplified movement and speed model was created (the simplification was carried out within reasonable limits). The next step is to develop a practical positioning strategy. After that, the elements and environment required for optimization were selected.

During the GA method testing, inconsistencies arose that caused the model to be subjected to several manual tunings (the main reason is the difference between factory and real drone parameters and not necessarily smooth movement, which could be due to several reasons, such as temperature and stock saturation issues). These contradictions, since they required relatively small modifications, can be easily eliminated.

In summary, in this section, the following necessary elements were found to help specify the method for creating the model:

- data, included in tables
    - ○ the fixed basic data,
    - ○ of the warehouse,
    - ○ of the drone,
    - ○ of the drone’s positioning within the warehouse
- the variable, i.e., the data to be searched as a target;
- the indexes used in bound;
- function specifying the speed of the drone;
- the processing route and time of the drone;
- the constraints, and
- the solving algorithm.

3.2. Basic Concepts

Compartment: a well-defined storage area on all sides, in which one or more goods or unit loads can be stored at the same time. The compartment has its own identifier. The compartment is rectangular.

Shelf system: a set of interconnected compartments used for storage, which are also connected to each other in their physical structure. In shelf systems, the compartments located above each other have the same width and depth. Their heights can vary, but the shelf system is rectangular.

Aisle: refers to roads suitable for transport between shelf systems. They can be two-way or one-way.

Charging aisle: the main road where the drone is located.

Main entrance: the place where the inventory starts and also ends.

Docking station charger: the non-operational location of the drone where the drone is charged.

3.3. The Known Data

In this section, known data related to the mathematical model are provided. The known data are the most important parameters of the warehouse and the drone: these are specifically provided by the storage company; the parameters of the drones are included in the technical description. The following tables summarize the data used for the model and provide the notations. Each index was used consistently for the same data, and an index table is also provided.

Agreements

In some warehouses, there is only one-way traffic in the aisles. One may enter one of the aisles and come back in the aisle(s) next to it. The following restriction does not constitute an actual restriction, but is for the sake of convenient handling. One can always enter from the side of the main entrance in case of an odd number of aisles and can come back on the even-numbered ones.

The selection of the main entrance is also arbitrary, but this does not affect the course of the solution either since the other side of the shelf systems is treated similarly. The main entrance was fixed from the point of view of the docking stations. The main entrance is where the docking stations were located. If there was a docking station on both sides, we could choose based on the floor plan, usage practice, or freely.

All drones took off at the same time. If this was not the case, but each drone started at a time different from the start time, then a time variable Δt_b could be assigned to each drone, which shows the time elapsed from the start of the inventory to the start of the drone. By adding this to T_f (see later), we obtained the time used by the given drone from the beginning of the inventory to the end of the processing. This does not affect the algorithm described below; in addition, this case occurs very rarely in practice, and due to the abovementioned reasons, this parameter can be omitted without restricting the generality.

The next table shows the indexes (Table 2).

Table 2. Table of indexes.

Indexes	Description
k, k_1	the serial number of the aisle.
$i, i_1,$	represents the rows of the shelf system.
$j, j_1,$	represents the columns of the shelf system.
β, β_1	position of the shelf system (left or right).
v	represents the sub-index of consecutive compartments.
u	index of charges.
b	index of drones.

The following figures (Figures 3–5) show the structure and parameters of the warehouse to be modeled.

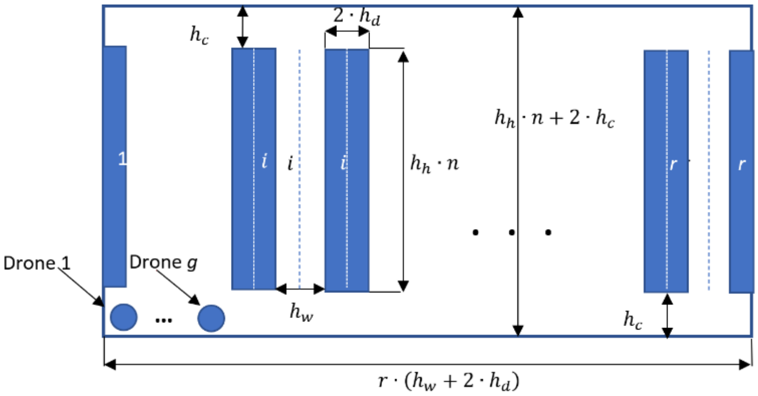

Figure 3. Floor plan of the warehouse, with dimensions included.

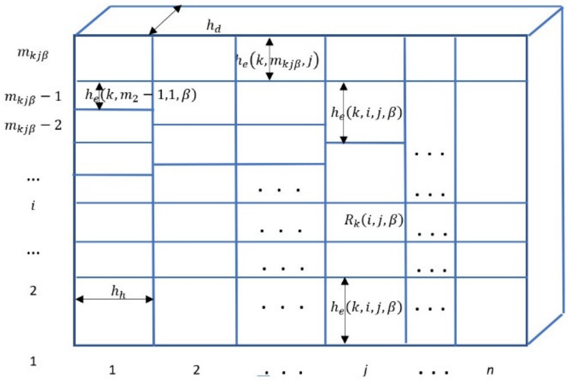

Figure 4. The data of the shelf system.

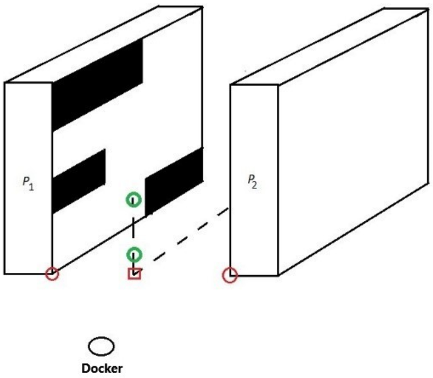

Figure 5. The shelf system to be tested. The black shelves indicate that the compartments that have already been examined. (The green circle indicates the processing starting point of the drone $(\mathbf{C}_{kb})$, the red circles the reference points of the shelf systems $(\mathbf{P}_l)$, the red square indicates the floor-level reference point of the aisles of two shelf systems, which is the same as $(\mathbf{C}_{kb})$, but $c_{ibz} = 0$.).

3.4. Variable Data

In this section, variable data are provided related to the mathematical model.

Denote by $\hat{m}$ the number of all bins, i.e., formally

$$\hat{m} = \sum_{k=1}^{r} \sum_{\beta=0}^{1} \sum_{i=1}^{n} \sum_{j=1}^{m_{ki\beta}} 1. \tag{1}$$

let $\hat{m}_b$ denote the number of compartments to be processed by the b-th drone:

$$\sum_{b=1}^{g} \hat{m}_b = \hat{m}. \tag{2}$$

For every drone, we need a

$$S_b = S_b\big((k_1 i_1 j_1 \beta_1), (k_2 i_2 j_2 \beta_2), \ldots, (k_{\hat{m}_b} i_{\hat{m}_b} j_{\hat{m}_b} \beta_{\hat{m}_b})\big)\ b = 1, \ldots, g \tag{3}$$

sequence of compartments, where S_b is belonging to all drones $\left(k_{l_1} i_{l_1} j_{l_1} \beta_{l_1}\right) \neq \left(k_{l_2} i_{l_2} j_{l_2} \beta_{l_2}\right)$ if $l_1 \neq l_2$.

All drones visited all compartments of all shelving systems while working. The compartment series S_b describes the sequence of these compartments visited by a drone, where the first parameter is the number of the shelf system, the other two are the number of shelves in the shelf system and the number of compartments within it, and the last one is whether it is examining the left or right shelf system of the aisle.

The next table shows the variable data (Table 3).

Table 3. Variable data table.

Variables	Description
$\mathbf{x} = \left[x_{k,i,j,\beta}\right]$	The sequence number of the processing (as the drones move between each compartment). The S_b-s belonging to each drone can be built from this.
$\mathbf{y} = \left[y_{k,i,j,\beta}\right]$	$y_{k,i,j,\beta} = b$ if the b-th drone processes the $R_k = R_k(i,j,\beta)$ compartment.
$T_f = T_f(S_b)$	The execution time of visiting the sequence of compartments S_b.
$t_P = t_P\left(\acute{S}_b\right)$	The $\acute{S}_b$ subseries execution time without charging.
$t_{T,R_k(i_p,j_p,\beta_p)}$	The travel time from the docking station to the $\left(i_p, j_p, \beta\right)$ starting point.
$t_{R_k(i_q,j_q,\beta_q),T}$	The travel time from the point $\left(i_q, j_q, \beta_q\right)$ to the docking station.
$t_{R_{k_1}(i_1,j_1,\beta_1),R_{k_2}(i_2,j_2,\beta_2)}$	The travel time from one compartment of a aisle to another compartment of the aisle.
$T_{R_{k_1}(i_1,j_1,\beta_1),R_{k_2}(i_2,j_2,\beta_2)}(C)$	The time from one compartment of one aisle to the compartment of the other aisle with a change at the beginning of the aisle.
$T_{R_{k_1}(i_1,j_1,\beta_1),R_{k_2}(i_2,j_2,\beta_2)}\left(\overline{\mathrm{C}}\right)$	The time from one compartment of one aisle to the compartment of the other aisle with a change at the end of the aisle.
p_b	The number of charges of the b-th drone-1.
$\hat{m}_b$	The number of compartments to be processed by the b-th drone.

Note: in the following, the *b*-th drone is always being examined in general, but all other drones can be described similarly.

The drones need to be charged from time to time. They may not be able to process the compartments assigned to them in one charge. For this reason, the compartment series of S_b of the b-th drone must be broken down into parts that can be processed with one charge, i.e., it contains the compartments processed during the operating time. Thus, the processing time consists of three parts:

- The drone moves to the first compartment to be processed,
- Processes compartments,
- Returns to the docking station.

The drone returns to the docking station when it comes to a compartment in which the available operating time for the previous compartment is still greater than Δt, but in the case of the current compartment it is already less, i.e., the operating time falls below a critical level. In addition, the individual subchains must output the entire chain, since each compartment must be examined in the order of S_b.

Let us denote the series of compartment tests carried out in one aisle at a time

$$\breve{S}_b = \breve{S}_b\left(\left(k_p i_p j_p \beta_p\right), \ldots, \left(k_q i_q j_q \beta_q\right)\right) \subset S_b \ (q \geq p) \tag{4}$$

if a shelf system needs to be changed, it is changed either in the direction of $\overline{\mathrm{C}}$ or C during the change.

Denote in $\breve{S}_b$ G the aisle change, i.e.,

$$G = \begin{cases} \left(k_p, k_{p+1}, 0\right), & \text{if it changes in direction } C \\ \left(k_p, k_{p+1}, -1\right), & \text{if it changes in direction } \overline{\mathrm{C}} \end{cases} \tag{5}$$

Denote

$$\acute{S}_b = \acute{S}_b\left(\breve{S}_{b1}, G_1, \breve{S}_{b2}, \ldots, G_{f-1}, \breve{S}_{bf}\right) \tag{6}$$

The division above is important during the practical examination. When solving the problem related to the model, the compartments examined during one charge is treated in a uniform manner; based on this, the time of $\acute{S}_b$ is also included in the compartment change, so $\breve{S}_{bi}$ is not needed in the calculations from now on. At the same time, if the optimal solution is given, it is necessary to know in which direction we are switching. $\breve{S}_{bi}$ must be entered here. Then, the execution time of the sequence $\acute{S}_b$ is from charger to charger

$$T_f\left(\acute{S}_b\right) = t_{T,R_k(i_p,j_p,\beta_p)} + t_P\left(\acute{S}_b\right) + t_{R_k(i_q,j_q,\beta_q),T} \tag{7}$$

if it is true

$$0 \leq T_w - T_f\left(\acute{S}_b\right) \leq \Delta t \tag{8}$$

Let

$$S_b = S_b\big((k_1 i_1 j_1 \beta_1), (k_2 i_2 j_2 \beta_2), \ldots, (k_{\hat{m}_b} i_{\hat{m}_b} j_{\hat{m}_b} \beta_{\hat{m}_b})\big) = \sum_{u=1}^{p} \acute{S}_{bu} \tag{9}$$

where every $\acute{S}_{bu}$

$$0 \leq T_w - T_f\left(\acute{S}_{bu}\right) \leq \Delta t \ (u = 1, \ldots, p-1). \tag{10}$$

This condition no longer has to be met for the last one.

The final access list should include a flight to the docking station. For this reason, S_b must be supplemented with the charging of the drone D in the appropriate places. The location of this is determined by the later algorithm (algorithm for generating objective function). The function $T_f\left(\acute{S}_{bu}\right)$ represents the operating time of the journey of each partial series from the docking station through processing to the docking station. The sum of these yields the total operating time. $T_f\left(\acute{S}_{bu}\right)$ is defined below.

Then, the goal is to minimize the processing time of the drone with the longest processing time. This also results in an even distribution of work between the drones:

$$T = \underset{b \in \{1,\ldots,g\}}{max} \sum_{u=1}^{p} T_f\left(\acute{S}_{bu}\right) \to min! \tag{11}$$

If T is minimal, then S_{bv} is the optimal route.

This is the objective function of the mathematical model.

3.5. The Speed of the Drone

The speed of AS drone is determined by knowing its horizontal travel speed and its vertical travel speed by projecting it onto the diagonal path. The red arrow indicates the velocity as a vector and its magnitude is the magnitude of the current velocity.

Height function

Let h_s denote the height difference between compartments $R_k(i_1, j_1\beta_1)$ and $R_k(i_2, j_2\beta_2)$ within the same shelf system:
if $i_1 > 0$ and $i_2 > 0$

$$\begin{gathered} h_s(k, i_1, j_1, \beta_1, i_2, j_2\beta_2) = \\ \left(\left(\sum_{i=1}^{i_2-1} h_e(k, i, j_2, \beta_2) + \tfrac{h_e(k, i_2, j_2, \beta_2)}{2}\right) - \sum_{i=1}^{i_1-1} h_e(k, i, j_1, \beta_1) + \tfrac{h_e(k, i_1, j_1\beta_1)}{2}\right). \end{gathered} \tag{12}$$

If $i_1 < 0$, then $-i_1$ shows the serial number of the drone. In this case, we consider the aisle-change height of drone $-i_1$ during the calculation:

$$h_s(k, i_1, j_1, \beta_1, i_2, j_2\beta_2) = \left(\sum_{i=1}^{i_2-1} h_e(k, i, j_2, \beta_2) + \frac{h_e(k, i_2, j_2, \beta_2)}{2}\right) - h_{-i_1}. \tag{13}$$

if $i_2 < 0$, then $-i_2$ shows the serial number of the drone. In this case, we consider the aisle-change height of drone $-i_2$ during the calculation:

$$h_s(k, i_1, j_1, \beta_1, i_2, j_2\beta_2) = h_{-i_2} - \left(\sum_{i=1}^{i_1-1} h_e(k, i, j_1, \beta_1) + \frac{h_e(k, i_1, j_{1,}\beta_1)}{2} \right). \tag{14}$$

More cases do not occur in practice.

Let x denote the horizontally mapped path of the drone's path and y the vertically mapped path of the drone's path (Figure 6).

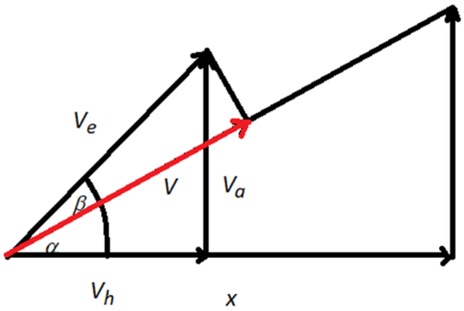

Figure 6. The speed of the drone.

When moving upwards

$$V_e = \sqrt{v_h^2 + v_a^2} \tag{15}$$

$$\alpha = \arctan\frac{y}{x},\ \beta = \arctan\frac{v_a}{v_h}. \tag{16}$$

When moving down

$$V_e = \sqrt{v_h^2 + v_d^2} \tag{17}$$

$$\beta = \arctan\frac{v_d}{v_h}. \tag{18}$$

It follows that the speed of the drone when moving up:

$$V_1(x, y) = V_e \cdot \cos(\beta - \alpha) = V_e \cdot \cos\left(\arctan\frac{v_a}{v_h} - \arctan\frac{abs(y)}{abs(x)}\right). \tag{19}$$

It follows that the speed of the drone when moving down:

$$V_2(x, y) = V_e \cdot \cos\left(\arctan\frac{v_d}{v_h} - \arctan\frac{abs(y)}{abs(x)}\right). \tag{20}$$

Based on these

$$V = V(x, y) = \begin{cases} y \geq 0 \text{ and } x \neq 0\ then\ V_2(x, y) \\ y < 0\ and\ x \neq 0\ then\ V_1(x, y) \\ y \geq 0 \text{ and } x = 0\ then\ V_a \\ y < 0\ and\ x = 0\ then\ V_d \end{cases} \tag{21}$$

It depends on the lengths of the road to be covered in the horizontal and vertical direction.

3.6. The Length and Time of Each Route

The drone starts all processing by being positioned in the middle of the two shelf systems in the green square position shown in Figure 2 ($\mathbf{C}_{kb}$). This means the center of the bottom row (it can be modified if necessary). The drone can take the best shot from the center of the compartments. This is confirmed by practical tests. The abovementioned periods are detailed in this section (Figure 7).

- The drone moves to the first compartment to be processed,
- Processes the QR codes of the compartments,
- Returns to the docking station.

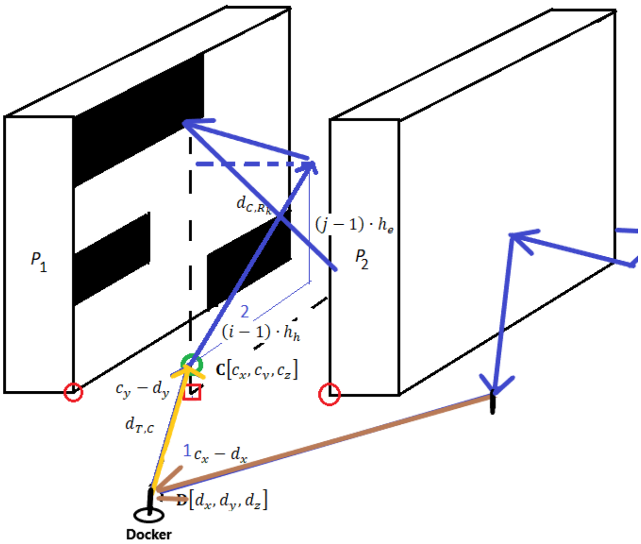

Figure 7. Processing process for individual drones.

3.6.1. The Distance and Time from the Docking Station to the Starting Point (Figure 8)

This consists of three parts:

I. The drone moves from the charger to the starting point between the shelves (the green view, $\mathbf{C}_{kb}$),

II. Then, it moves from here to the first compartment to be processed,

III. Then, it rotates the camera in the appropriate direction.

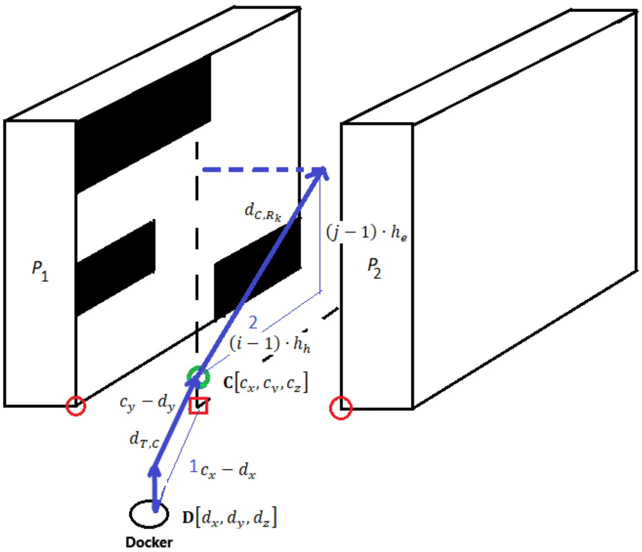

Figure 8. The route and the auxiliary lines helping to determine it. (The blue line shows the drone's path from the docking station to the first compartment to be processed).

I. Time to reach the starting position of the shelves:

$R_k = R_k(i, j, \beta)$ is the compartment distance from the docking station:

$$d_{T,R_k(i,j,\beta)} = d_{T,C_{kb}} + d_{C_{kb},R_k}. \tag{22}$$

$R_k = R_k(i, j, \beta)$ is the compartment access time from the docking station:

$$t_{T,R_k(i,j,\beta)} = t_{T,C_{kb}b} + t_{C_{kb}b,R_k}, \tag{23}$$

where d_{T,C_k} is the path between the docking station and the starting position of the middle guide path between the two shelves to the green square (in the case of the f-th shelf system, where $\frac{h_h}{2}$ is added because it always stands in the middle of the starting compartment):

$$d_{T,C_{kb}} = h_b + \sqrt{(c_{kbx} - d_{bx})^2 + \left(c_{kby} - d_{by}\right)^2} + \frac{h_h}{2} \tag{24}$$

t_{T,C_f} is the time required to reach the starting position of the middle guideway between the docking station and the two shelves:

$$t_{T,C_{kb}} = \frac{h_b}{V_a} + \frac{\sqrt{(c_{kbx} - d_{bx})^2 + \left(c_{kby} - d_{by}\right)^2} + \frac{h_h}{2}}{V_h}. \tag{25}$$

II. Distance and time from the center to the starting compartment:

$$d_{C_{kb},R_k} = d_{R_k,C_{kb}} = \sqrt{((j-1)\cdot h_h)^2 + h_s^2(k, i, j, \beta, -b, 1, \beta)} \tag{26}$$

$$t_{C_{kb},R_k} = \frac{d_{C_k,R_k}}{V((j-1)\cdot h_h, h_s(k, i, j, \beta, -b, 1, \beta))} + \frac{90}{V_r} + T_{PH} \tag{27}$$

III. The camera
$\frac{90}{V_r}$ is the camera rotation time. Travel time to the starting position:

$$\begin{gathered} t_{T,R_k(i,j,\beta)} = t_{T,C_{kb}} + t_{C_{kb},R_k} = \\ \frac{h_b}{V_a} + \frac{\sqrt{(c_{kbx}-d_x)^2+\left(c_{kby}-d_y\right)^2}+\frac{h_h}{2}}{V_h} + \\ \frac{\sqrt{((j-1)\cdot h_h)^2+h_s^2(k,i,j,\beta,-b,1,\beta)}}{V((j-1)\cdot h_h,h_s(k,i,j,\beta,-b,1,\beta))} + \frac{90}{V_r} + T_{PH}. \end{gathered} \tag{28}$$

3.6.2. The Time to Reach the Docking Station from a Given Point Can Be Determined Similarly

This way is the reverse, since we

- Rotate the camera to the starting position,
- Move to the starting point of the shelf system according to the given drone,
- Return to the docking station.

Similarly:

$$t_{R_k,C_{kb}} = \frac{d_{R_k,C_k}}{V\left(h_s(k, i, j, \beta, -b, 1, \beta), \left(j - \frac{1}{2}\right)\cdot h_h\right)} + \frac{90}{V_r} + T_{PH} \tag{29}$$

$$\begin{gathered} t_{R_k(i,j,\beta),T} = t_{R_k,C_{kb}} + t_{C_{kb},T} = \\ \frac{90}{V_r} + \frac{\sqrt{\left(\left(j-\frac{1}{2}\right)\cdot h_h\right)^2+h_s^2(k,i,j,\beta,-b,1,\beta)}}{V\left(\left(j-\frac{1}{2}\right)\cdot h_h,h_s(k,i,j,-b,1)\right)} + \\ \sqrt{\frac{(c_{kx}-d_x)^2+\left(c_{ky}-d_y\right)^2}{V_h^2}} + \frac{h_b}{2V_d}. \end{gathered} \tag{30}$$

3.6.3. A Path between Two Compartments within the Same Aisle

The path and time between the next two compartments in the processing line were determined (this is not affected by the fact that the two compartments were opposite to each other). For the time being, it is enough to calculate the speed and divide the distance by this to obtain the time. In addition, the camera could be rotated if necessary (by 180°). The rotation of the camera as needed was controlled by the β value of the two apertures. The time for photography was added at the end (Figure 9).

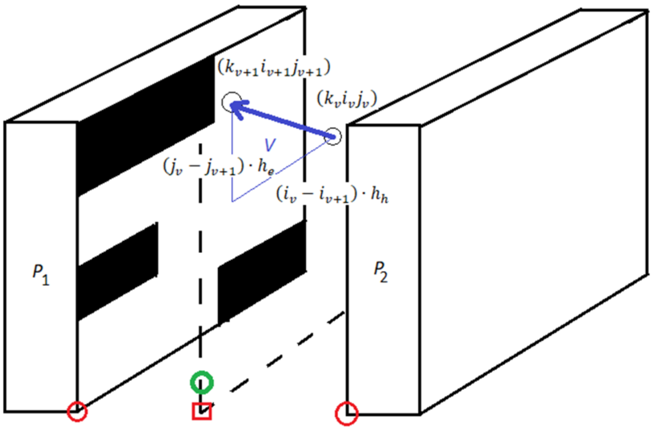
$(k_{v+1}i_{v+1}j_{v+1})$
$(k_v i_v j_v)$
V
$(j_v - j_{v+1}) \cdot h_e$
$(i_v - i_{v+1}) \cdot h_h$
P_1
P_2

Figure 9. The path from compartment to compartment within an aisle. Two shelf systems appear here since we observe two at the same time. (The blue line indicates the path of the drone from the processed compartment to the next compartment).

Let $(k_v i_v j_v)$, $(k_{v+1}i_{v+1}j_{v+1})$ to be two points, where $k_v = k_{v+1}$. Their distance is:

$$d((k_v i_v j_v, \beta_v), (k_v i_{v+1} j_{v+1}, \beta_{v+1})) = \sqrt{((j_v - j_{v+1}) \cdot h_h)^2 + h_s^2(k, i_v, j_v, \beta_v, i_{v+1} j_{v+1}, \beta_{v+1})}. \tag{31}$$

Time spent by the drone on this route:

$$t_{P_1}((k_v i_v j_v, \beta_v), (k_{v+1} i_{v+1} j_{v+1}, \beta_{v+1})) = \frac{\sqrt{((j_v - j_{v+1}) \cdot h_h)^2 + h_s^2(k, i_v, j_v, \beta_v, i_{v+1} j_{v+1}, \beta_{v+1})}}{V((j_v - j_{v+1}) \cdot h_h, h_s(k, i_v, j_v, \beta_v, i_{v+1} j_{v+1}, \beta_{v+1}))} + \frac{abs(\beta_v - \beta_{v+1}) \cdot 180}{V_r} + T_{PH} \tag{32}$$

where

$$\frac{abs(\beta_v - \beta_{v+1}) \cdot 180}{V_r} \tag{33}$$

is the camera rotation (i.e., it only rotates when the two apertures are on opposite sides).

3.6.4. From One Compartment of One Row to Another Compartment of Another Row

In this subsection, the path and processing time of the two consecutive compartments to be processed, which are located in different rows, are given (Figure 10).

- The drone rotates the camera to the starting position and moves to the start or end point of the shelf system according to the drone performing the processing (C, or $\overline{C}$) at the given aisle height,
- Then, it moves to the start or end point (C, vagy $\overline{C}$) of the next shelf system to be tested according to the drone at the given aisle height.
- Then, it returns to the starting slot of the row.

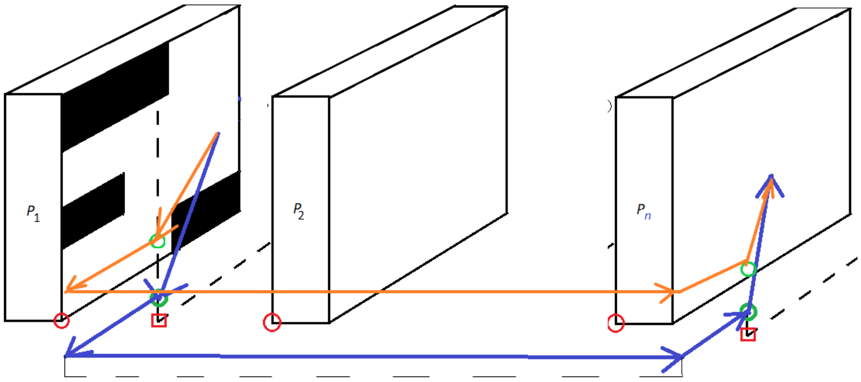

Figure 10. Movement from compartment to compartment in the case of different aisles. The figure shows the changeover at the starting point of the aisle, but the situation is similar if the changeover takes place at the other end of the shelf system. The orange path shows the path of another drone. It has a different route to avoid collisions.

Let us denote

$$a(j) = \begin{cases} j-1, & \text{if } C \text{ is the examined point} \\ n-j, & \text{if } \overline{C} \text{ is the examined point} \end{cases} \tag{34}$$

This ensures that aisles can be changed at both ends of the shelving system.

To change aisles, the drone must move from the start or end point of the shelf system to the middle of the aisle. This is given by $\frac{h_c}{2h_h}$.

$$\begin{gathered} T_{R_{k_1}(i_1,j_1,\beta_1),R_{k_2}(i_2,j_2,\beta_1)}(C) = t_{R_{k_1},C_{kb_1}} + \frac{h_c}{2h_h} + t_{C_{k_1},C_{kb_2}} + \frac{h_c}{2h_h} + t_{C_{kb_1},R_{k_1}} = \\ \frac{d_{R_{k_1},C_{kb_1}}}{V(a(j_1)\cdot h_h, h_s(k_1,i_1,j_1,\beta_1,-b,a(j_1),\beta_1))} + \frac{90}{V_r} + \\ \frac{\frac{h_c}{2} + \text{abs}(k_1-k_2)\cdot(h_w+2h_d) + \frac{h_c}{2}}{v_h} + \\ \frac{d_{C_{kb_2},R_{k_2}}}{V(h_s(k_2,i_2,j_2,\beta_2,-b,a(j_2),\beta_2),a(j_2)\cdot h_h)} = \\ \frac{d_{R_{k_1},C_{kb_1}}}{V(a(j_1)\cdot h_h, h_s(k_1,i_1,j_1,\beta_1,-b,a(j_1),\beta_1))} + \frac{h_c + \text{abs}(k_1-k_2)\cdot(h_w+2h_d)}{v_h} + \\ \frac{d_{C_{k_2},R_{k_2}}}{V(h_s(k_2,i_2,j_2,\beta_2,-b,a(j_2),\beta_2)a(j_2)\cdot h_h)} + \frac{90}{V_r} + T_{PH}. \end{gathered} \tag{35}$$

Similarly to the optimal path, we calculate

$$T_{R_{k_1}(i_1,j_1,\beta_1),R_{k_2}(i_2,j_2,\beta_2)}\left(\overline{C}\right) = t_{R_{k_1},\overline{C}_{k_1}} + t_{\overline{C}_{k_1},\overline{C}_{k_2}} + t_{\overline{C}_{k_1},R_{k_1}}. \tag{36}$$

Two cases occur:

(a) The case of progress without direction

$$t_{R_{k_1}(i_1,j_1,\beta_1),R_{k_2}(i_2,j_2,\beta_2)} = min\left(T_{R_{k_1}(i_1,j_1,\beta_1),R_{k_2}(i_2,j_2,\beta_2)}(C), T_{R_{k_1}(i_1,j_1,\beta_1),R_{k_2}(i_2,j_2,\beta_2)}\left(\overline{C}\right)\right). \tag{37}$$

(b) Aisle change in the case of one-way traffic

Several warehouses have one-way traffic. The direction of travel is the opposite in odd-numbered and even-numbered aisles. Then

- if k_1 is odd, then k_2 is even and the point of change is $\overline{C}$ and $t_{R_{k_1}(i_1,j_1,\beta_1),R_{k_2}(i_2,j_2,\beta_2)} = T_{R_{k_1}(i_1,j_1,\beta_1),R_{k_2}(i_2,j_2,\beta_2)}\left(\overline{C}\right)$
- if k_1 is even, then k_2 is odd and the point of change is C and $t_{R_{k_1}(i_1,j_1,\beta_1),R_{k_2}(i_2,j_2,\beta_2)} = T_{R_{k_1}(i_1,j_1,\beta_1),R_{k_2}(i_2,j_2,\beta_2)}(C)$.

Considering that there may be two compartments to be examined one after the other within an aisle or between different aisles, the t_P function can be of two types, of which only the current one is always included:

$$t_P((k_v i_v j_v \beta_v), (k_{v+1} i_{v+1} j_{v+1} \beta_{v+1})) = \begin{cases} t_{P_1}((k_v i_v j_v \beta_v), (k_{v+1} i_{v+1} j_{v+1} \beta_{v+1})) \text{ if } k_{v+1} = k_v \\ t_{R_{k_v}(i_v, j_v, \beta_v), R_{k_{v+1}}(i_{v+1}, j_{v+1}, \beta_{v+1})} \text{ if } k_{v+1} \neq k_v \end{cases} \tag{38}$$

The $\acute{S}_{bu}$'s subsequent (dockless) execution time

$$t_P(\acute{S}_{bu}) = \sum_{v=p}^{q-1} t_P((k_v i_v j_v \beta_v), (k_{v+1} i_{v+1} j_{v+1} \beta_{v+1})). \tag{39}$$

$$T_f(\acute{S}_{bu}) = t_{T, R_k(i,j,\beta)} + t_P(\acute{S}_{bu}) + t_{R_k(i,j,\beta), T}. \tag{40}$$

3.7. The Constraints

Based on the abovementioned, the sum of the compartment-order variables is exactly the sum of the number of cells from 0 to 1 (starting with 0) according to the number of cells.

During the condition set of the model, we must ensure four plus two conditions:

1. All compartment serial numbers must be different (this should only be carried out for compartments belonging to a drone, but from the point of view of the solution, it is easier to handle it this way, and at the same time, it does not affect the final result).
2. The sequence numbers should include all values starting from 0 up to all compartment numbers (i.e., the sequence numbers should increase one by one).
3. Each compartment must be assigned to exactly one drone.
4. Only one drone can process an aisle.

Note. In the case of one-way traffic when changing aisles, two additional conditions apply:

5. For each drone, the first compartment to be processed can be in an odd-numbered aisle.
6. In the event of changing aisles during processing, the sequence numbers of the two aisles must have different parities.

The first and third conditions can be specified as group conditions directly in the software in which the problem is run.

Condition 1 and condition 2

$x_{kij\beta}$ cannot be negative (because it is a traversal sequence); for this reason,

$$0 \leq x_{kij\beta}. \tag{41}$$

The value of x_{kij} can be clearly limited from the above as follows:

$$x_{kij\beta} \leq \hat{m}. \tag{42}$$

Each number must be different if different compartments are being examined, as follows:

$$k, k_1 \in [1, \ldots, r], j,\ j_1 \in [1, \ldots, n],\ i \in \left[1, \ldots, m_{k,j}\right], i_1 \in \left[1, \ldots, m_{k_1 j_1}\right], \beta, \beta_2 \in [0, 1] \tag{43}$$

then,

$$\begin{gathered} 10000k + 100i + 10j + \beta \neq 10000k_1 + 100i_1 + 10j_1 + \beta_1, \\ x_{kij\beta} \neq x_{k_1, i_1, j_1 \beta_1}. \end{gathered} \tag{44}$$

Condition 3

The condition for assigning drones to compartments is calculated as follows:

$$1 \leq y_{kij\beta} \leq\ g,\ y_{kij\beta} = int \tag{45}$$

i.e., the processing of each compartment belongs to exactly one drone.

Condition 4

Two drones cannot be in the same aisle at the same inventory, so

$$\begin{gathered} k, k_1 \in [1, \ldots, r], j,\ j_1 \in [1, \ldots, n],\ i \in \left[1, \ldots, m_{k,j}\right], i_1 \in \left[1, \ldots, m_{k_1 j_1}\right],\ \beta, \beta_1 \in [0,1] \\ k = k_1 \text{ then } y_{kij\beta} = y_{k_1 i_1 j_1 \beta_1} \end{gathered} \tag{46}$$

Condition 5 and condition 6

Aisles change in case of one-way traffic

In some warehouses, for safety reasons, the aisle between each shelving system can only be used in one direction. Usually, two adjacent aisles are reversed so that at the end of the aisle one can return down the aisle immediately adjacent to it. This implies that every second corridor has the same direction of travel, so we can assume that the direction of travel in even-numbered and odd-numbered corridors is the same.

For this, the model must be supplemented with new conditions. Let

$$Z_b = \left\langle z_f = x_{kij\beta} \middle| y_{kij\beta} = b \right\rangle \tag{47}$$

be a set ordered in increasing value according to $x_{kij\beta}$ (it has significance from the implementation side, not in the model). Then, the starting condition is that each drone can only start in an odd aisle (due to condition 1). The restriction of changing aisles appears when calculating $T_f\left(\acute{S}_{bu}\right)$; therefore:

$$\operatorname{mod}\left(\hat{k}_b, 2\right) = 1, \text{where } x_{\hat{k}_b \hat{i}_b \hat{j}_b \hat{\beta}_b} = \min_{x_{kij\beta} \in Z_b} x_{kij\beta}. \tag{48}$$

The second condition is as follows:

$$\text{abs}(\text{mod}(k, 2) - m\text{od}(k_1, 2)) = 1, x_{kij\beta}, x_{k_1 i_1 j_1 \beta_1} \in Z_b,\ k \neq k_1, \tag{49}$$

if it is true, then

$$\begin{gathered} x_{k_2 i_2 j_2 \beta_2} \neq x_{kij\beta},\ x_{k_2 i_2 j_2 \beta_2} \neq x_{k_1 i_1 j_1 \beta_1},\ x_{k_2 i_2 j_2 \beta_2} \in X_b, \\ x_{k_2 i_2 j_2 \beta_2} < min\left(x_{kij\beta}, x_{k_1 i_1 j_1 \beta_1}\right), \text{ or } x_{k_2 i_2 j_2 \beta_2} > max\left(x_{kij\beta}, x_{k_1 i_1 j_1 \beta_1}\right). \end{gathered} \tag{50}$$

The latter means indicates if the aisle numbers of the two compartments to be processed consecutively are not equal, the direction of the aisle of the first compartment to be processed must be opposite to the direction of the corridor of the next compartment.

3.8. Summary of the Model

$$\begin{gathered} k, k_1 \in [1, \ldots, r], j,\ j_1 \in [1, \ldots, n],\ i \in \left[1, \ldots, m_{k,j}\right], i_1 \in \left[1, \ldots, m_{k_1 j_1}\right],\ \beta, \beta_1 \in [0,1] \\ x_{kij\beta} = int,\ y_{kij\beta} = int \\ 0 \leq x_{kij\beta} \leq\ \hat{m} \\ 1 \leq y_{kij\beta} \leq\ g \\ k = k_1 \text{ then } y_{kij\beta} = y_{k_1 i_1 j_1 \beta_1} \\ \frac{x_{kij\beta} \neq x_{k_1, i_1, j_1 \beta_1}\ \ ha\ 10000k + 100i + 10j + \beta \neq 10000k_1 + 100i_1 + 10j_1 + \beta_1,}{T = \max\limits_{b \in \{1, \ldots, g\}} \sum\limits_{u=1}^{p} T_f\left(\acute{S}_{bu}\right) \rightarrow min!} \end{gathered} \tag{51}$$

3.9. Solution Method of the Task Related to the Model

The computational model can be further simplified if

$$0 \leq x_{kij\beta} \leq\ \hat{m} \tag{52}$$

instead of this condition, the following is used:

$$0 \leq x_{kij\beta} \leq mtp \cdot \hat{m} \tag{53}$$

(e.g., $mtp = 100$ can be). We introduce a condition and use the following routine as the non-linear limiting condition. This is important for the genetic algorithm. Then, it is sufficient that all values are different. This can then be transformed into an ordered list according to (41–42) with a simple auxiliary routine.

The name of the nonlinear limiting function is *Korfel_fv*. This satisfies condition (41–50). If it is fulfilled, it returns -1; if not, it returns 3.

Note. In the following we use the ":=" operation. Its meaning is different from the "=" sign. The meaning of ":=" is that the value of the variable on the left side of the operation is equal to the expression on the left side.

3.9.1. The Algorithm for Generating the Objective Function (Fitnesz_fv)

The algorithm specifies the algorithm for one drone (drone b), but the function in a similar way for all drones can be generated. The value of the objective function for the current allocation is given by T_{max}.

Step 1

Let

$$S = S((k_1 i_1 j_1 \beta_1), (k_2 i_2 j_2 \beta_2), \ldots, (k_{\hat{m}} i_{\hat{m}} j_{\hat{m}} \beta_{\hat{m}})) \tag{54}$$

be the compartment sequence where

$$\left(k_{l_1} i_{l_1} j_{l_1} \beta_{l_1}\right) \neq \left(k_{l_2} i_{l_2} j_{l_2} \beta_{l_2}\right) \text{ if } l_1 \neq l_2. \tag{55}$$

Each compartment should be included in the test once and only once. (When using the GA algorithm, this random generation is carried out for each element of the initial population).

We also have

$$\mathbf{y} = \left[y_{kij\beta}\right],\ 1 \leq y_{kij\beta} \leq\ g. \tag{56}$$

Let

$$T_{max} = 0 \text{ and } b = 1 \tag{57}$$

Step 2

Let $u = 0$, $S_{bv} = \varnothing$, and let be $e = (k_1 i_1 j_1 \beta_1)$ the first element of the sequence such that $y_{k_1 i_1 j_1 \beta_1} = b$ (if it exists).

Step 3

If there are no more elements of the series, continue from Step 5. If there are, let

$$\begin{gathered} u := u + 1 \\ S'_{bu} = \varnothing. \end{gathered} \tag{58}$$

Step 4

We add this element of the series to the subseries:

$$S'_{bu} := S'_{bu} + \{e\}, \tag{59}$$

Let us calculate the value of

$$T_f\left(S'_{bu}\right) \tag{60}$$

If

$$T_w - T_f\left(S'_{bu}\right) > \Delta t \tag{61}$$

then we take the next element e and continue from Step 4.

If

$$T_w - T_f\left(S'_{bu}\right) \leq 0 \tag{62}$$

i.e., if the drone would not return in time to receive the new element, then

$$\begin{gathered} S'_{bu} := S'_{bu} - \{e\} \\ T := T + T_f\left(S'_{bu}\right) \\ S_{bv} := S_{bv} + \left\{\mathbf{D}, S'_{bu}\right\} \end{gathered} \tag{63}$$

Then, continue from Step 3.

If

$$0 \le T_w - T_f\left(S'_{bu}\right) \le \Delta t \tag{64}$$

then, let

$$\begin{gathered} T := T + T_f\left(S'_{bu}\right) \\ S_{bv} := S_{bv} + \left\{\mathbf{D}, S'_{bu}\right\} \end{gathered} \tag{65}$$

then, take the next element g and continue from Step 3.

Step 5

Let

$$S_{bv} := S_{bv} + \{\mathbf{D}\} \tag{66}$$

At that time, the task list is created, completed with the charges, and the value of the total processing time T for the b-th drone is also received.

Step 6

If

$$T_{max} < T \text{ then let have } T_{max} = T \tag{67}$$

and

$$b := b + 1 \tag{68}$$

If $b \le g$, then continue from Step 2 to determine the function value.

3.9.2. Checking the Fulfilment of the Limiting Conditions (Korfel_fv)

Condition 2. The error shows whether the conditions are fulfilled (-1) or not fulfilled (3), which is its initial value:

$$Error = -1. \tag{69}$$

Each value in x must be different. It can be easily checked by pairwise comparison. If it is not fulfilled, then $Error = 3$ and stop.

Condition 4. An aisle can only be processed by a drone's condition algorithm.

Step 1

Let the value of k be the code of the first aisle, i.e.,

$$k = 1. \tag{70}$$

Step 2

Let

$$b = y_{k,1,1,0} \tag{71}$$

For each $y_{kij\beta}$ it is checked whether the condition $y_{kij\beta} = b$ is fulfilled for the same k value. If it is not fulfilled, then $Error = 3$ and stop the check.

Step 3

If at the end of the check we obtain a value of $Error = -1$, then

$$k := k + 1 \tag{72}$$

and if $k \le r$, then continue from Step 2.

Conditions 5 and 6. Check the algorithm for one-way traffic conditions (in case of two-way traffic, this test is omitted).

Step 1

Let the code of the first drone be b:

$$b = 1. \tag{73}$$

Step 2

$$x_{k_0 i_0 j_0 \beta_0} = \min_{y_{kij\beta}=b} x_{kij\beta} \tag{74}$$

We check if k_0 is odd. If not, then $Error = 3$ and stop the check.

If it is, then for every pair that belongs to two compartments is to be processed consecutively (i.e., $x_{k_l i_l j_l \beta_l}$, $x_{k_{l+1} i_{l+1} j_{l+1} \beta_{l+1}}$), there is no value between them in the line belonging to the given drone, and $y_{k_l i_l j_l \beta_l} = y_{k_{l+1} i_{l+1} j_{l+1} \beta_{l+1}}$ is checked for parity. If k_l and k_{l+1} are even at the same time, or if ha k_l and k_{l+1} are odd at the same time, $Error = 3$ and stop.

Step 3

If $Error = 3$, then

$$b := b + 1. \tag{75}$$

If $b \leq g$, then continue from Step 2.

3.9.3. Determining the Optimal Access Route

An evolutionary algorithm is used for the solution.

Step 1

For setting the initial parameters (number of populations, number of steps, etc.), formation of the initial population (there is no longer any need to examine drones separately; it is enough to deal with the allocation of all compartments and the assignment of drones), i.e., each chromosome is

$$\mathbf{x} = \mathbf{x}((k_1 i_1 j_1 \beta_1), (k_2 i_2 j_2 \beta_2), \ldots, (k_{\hat{m}} i_{\hat{m}} j_{\hat{m}} \beta_{\hat{m}})) \tag{76}$$

and

$$\mathbf{y} = \left[y_{kij\beta} \right],\ 1 \leq y_{kij\beta} \leq\ g. \tag{77}$$

Step 2

Let us run the algorithm in a way that the fitness value is determined for each individual of the population with the help of the Objective function generation algorithm (Fitnesz_fv). Let us use the algorithm for checking the fulfilment of limiting conditions during the test (Korfel_fv). This can also be incorporated into factory algorithms (e.g., as a nonlinear limiting condition in MatLab).

Step 3

After the algorithm stopped, we evaluated the result and programmed the resulting optimal solution (x based on y) into the drone along with the necessary camera rotations and photography.

In the end, even for individual drones, the routes for all drones can be optimized, except for the drone with the longest execution path, one-by-one with this algorithm (narrowing x down to the compartments of drone b). It is easy to verify that the optimal solution is no longer affected by these steps.

4. Results and Discussion

Before applying the method to real warehouses, the usability of the model and the solution algorithm for a simplified, but real warehouse was examined. A drone was used during the investigation. The structure of the specific warehouse is as follows (Figures 11 and 12):

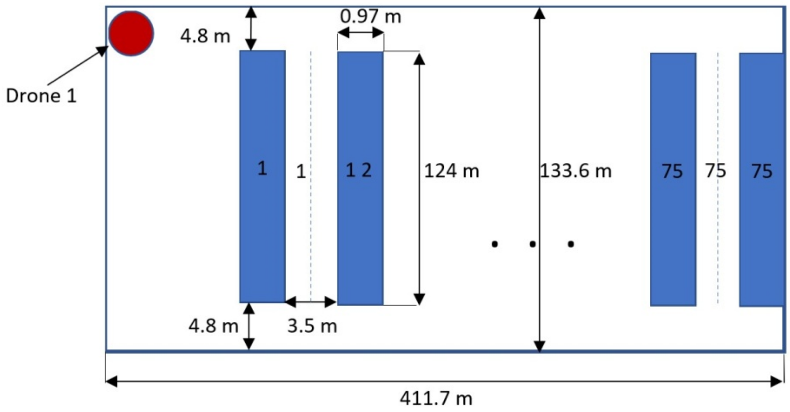

Figure 11. Floor plan of the warehouse, with dimensions included.

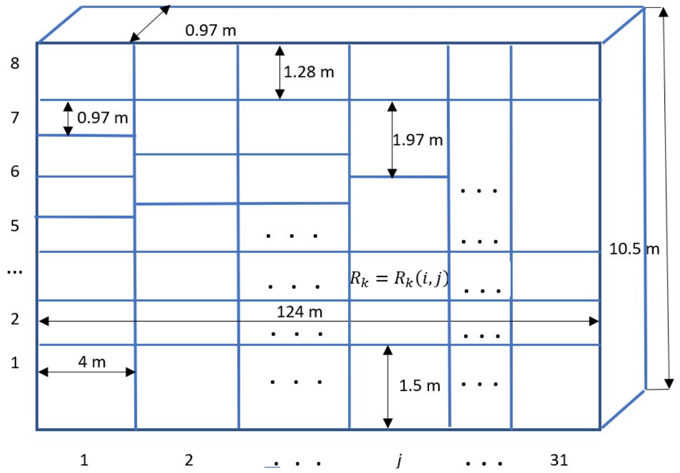

Figure 12. The data of the shelf system.

We ran the MatLab application supplemented with the algorithms given in the previous chapter with the following data (Table 4).

Table 4. The most important data of the drone.

Data of the drone	
$\mathbf{D} = \mathbf{D}[d_x, d_y, d_z]$	[0.5; 0.5; 0]
T_w	1380 s
T_{PH}	1 s
V_h	10 $m{\cdot}s^{-1}$/s
V_a	5 $m{\cdot}s^{-1}$
V_d	3 $m{\cdot}s^{-1}$
V_r	450 $degree{\cdot}s^{-1}$
Δt	50 s
T_{ch}	3600 s

The results are as follows:

1. The total number of possible cases was 11,780, which means a lot of cases, and that is why we chose a solution and control method that, if possible, quickly finds a good solution. It seems that a longer running time is needed, since in this special case the chromosome contains 11,780 genes. For this reason, the population has to be increased.

With the introduction of the variable mtp, the search for possible solutions has become easier, since the sequence numbers can not only be narrowly selected from the values $1, \ldots, r{\cdot}\hat{m}{\cdot}n$, but from $1, \ldots, mtp{\cdot}r{\cdot}\hat{m}{\cdot}n$. However, this requires a conversion routine, which at the end of the run converts the received vector x back into serial numbers.

2. The algorithm was tested with several speed values. It is obvious that, in the case of the current warehouse structure, $(h_e = 2.1,\ h_h = 4)$; then, if

$$V_a, V_d \ll V_h, \tag{78}$$

the walk-in is carried out aisle by aisle, processing the compartments of the bottom row. Otherwise, the drone's program will use column traversal, also starting from the bottom row. It is also concluded that the examination of the opposite compartments was carried out in sequence.

3. If (78) does not hold, then the genetic algorithm does not always provide a clear result. All the obtained results were close to the optimal value; however, the traversal was not carried out regularly as described above, but—in some cases—with randomly chosen compartments. It produces different results when run several times under the same conditions and parameters. For this reason, the parameters must be examined more closely and adjusted accordingly. Fortunately, this did not happenin practice with the examined drones.
4. In our specific case, condition (79) was fulfilled, so according to the solution, if the docking station is in front of the first aisle, then the shelf system close to the docking station starts processing with a pair, moves towards the end and turns back at the last shelf system. It was confirmed during the run that if the drone starts with a full charge (which is always the case in practice), then it does not need to be charged at the given warehouse on the way with this visit.

$$\frac{180^\circ}{V_r} < min\left(\frac{h_h}{V_h}, \frac{h_e}{V_a}, \frac{h_d}{V_d}\right) \tag{79}$$

5. The situation was similar when taking pictures. If the rotation speed of the camera was shorter than the speed of movement (horizontal, vertical) between the compartments (79), then the opposite compartments were photographed first; if not, then a shelf system was photographed first, and then the opposite shelf system was photographed afterwards.
6. During the test, several different docking positions were tested. The situation is not more complicated in these cases either. Even in this case, the drone started at the first shelf system. In this case, it depends on which end aisle the docking station is located, since the drone started with the first compartment of the first shelf system closest to the aisle.
7. It was confirmed that the model describes the practical problem well and is suitable for the general investigation of the problem. The solution method is a bit uncertain, but if condition (78) is fulfilled, it solves the task well and returns the inventory program in the shortest time.
8. In the future, we will continue to investigate possible solutions to see whether a more stable, less sensitive solution method can be found.

In this study, the authors developed a new model for the automatic inventory of large warehouses using drones. By examining several warehouses, the authors formulated a general mathematical model to describe the problem. In addition, the algorithms for solving the problem related to the model were given, thus enabling the practical application of the obtained results. At the same time, the effect of the drone setup in solving warehouse tasks was examined. This model and the solution of the related task enable the description of the impact of automatic warehouse processing with the help of drones on sustainability aspects, especially with regards to energy consumption.

The main added value of the article is that it showed that the problem can be modeled mathematically and thus, in general, the problem can be solved using this model. An additional added value is the findings that the use of drones has a significant impact on the automation, speed, and energy efficiency of warehouse processing, while considering all the task-relevant constraints associated with warehouses.

The scientific contribution of this article for researchers in the field of the use of drones in the case of warehouse processes is the mapping of the main elements of the problem, the generalization of previously examined elements in practice, the mathematical model of the new, unique approach, and the time functions that specify the optimal route of processing, as well as the solving algorithm and the related auxiliary algorithms. The new results also confirm the important basic hypothesis that the problem can be solved.

The results of this study can be generalized, since several elements were incorporated into the model (e.g., the use of several drones, with different start times, if necessary different shelves, one-way, multi-way traffic) that do not necessarily occur in specific warehouses, or only a part of them, as well as for all similar systems (which are not necessarily warehouses), allowing the model and solution to be applied in contexts in which our concepts and definitions fit. It follows that the optimal processing routes and management of steps can be used or applied to various other types of drone-based services.

The results of the research can help managerial decisions, as the described method enables the analysis of individual processes of automatic warehouse management and the support of strategic decisions by examining various parameters. In this way, a warehouse management mechanism can be selected that best serves the set logistics goals.

The limitations of the model include automatic handling of faulty aperture processing (imaging). Although this occurs in a very small percentage of cases, handling the issue provides further research opportunities. In addition, the stability of the solving method also raises questions (according to the first approach, it seems to be very parameter sensitive), so it is worth performing a sensitivity test and improving it with further research. It is also worth examining a wider range of solution methods, possibly making the solution algorithm more stable by combining it with other, newer methods.

5. Conclusions

In this paper, we presented a mathematical model and a solution method for a multi-user warehouse inventory with drones.

Our previous investigations and experiences showed that the given mathematical model describes most of the warehouses and the drone inventory task in practice well. This also shows that, in practice, the model can be used well in logistic decisions.

As described in the previous chapter, we encountered some problems in our solution method, which fortunately did not cause any problems for the warehouses we investigated, but further investigations are needed to ensure the stability of the solution.

Author Contributions: Conceptualization, M.G.; methodology, M.G.; writing—review and editing, J.U.; visualization, J.U.; investigation, M.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Acknowledgments: This research project was carried out within the framework of Centre of Excellence for Future Value Chains of Budapest Business School.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Rhiat, A.; Chalal, L.; Saadane, A. A Smart Warehouse Using Robots and Drone to Optimize Inventory Management. *Lect. Notes Netw. Syst.* **2022**, *358*, 475–483. [CrossRef]
2. Fülöp, M.T.; Udvaros, J.; Gubán, Á.; Sándor, Á. Development of Computational Thinking Using Microcontrollers Integrated into OOP (Object-Oriented Programming). *Sustainability* **2022**, *14*, 7218. [CrossRef]

3. Karamitsos, G.; Bechtsis, D.; Tsolakis, N.; Vlachos, D. Unmanned aerial vehicles for inventory listing. *Int. J. Bus. Syst. Res.* **2021**, *15*, 748–756. [CrossRef]
4. Bányai, T. Impact of the Integration of First-Mile and Last-Mile Drone-Based Operations from Trucks on Energy Efficiency and the Environment. *Drones* **2022**, *6*, 249. [CrossRef]
5. Francuz, Á.; Bányai, T. Branch and Bound Solution of Routing Problems for Drone-Based Supply. *Acad. J. Manuf. Eng.* **2022**, *20*, 20–26.
6. Agárdi, A.; Kovács, L.; Bányai, T. Vehicle Routing in Drone-Based Package Delivery Services. In *Solutions for Sustainable Development*, 1st ed.; Szita Tóthné, K., Jármai, K., Voith, Eds.; CRC Press: London, UK, 2019; Volume 1, pp. 151–159. [CrossRef]
7. Khan, A.A.; Laghari, A.A.; Gadekallu, T.R.; Shaikh, Z.A.; Javed, A.R.; Rashid, M.; Estrela, V.V.; Mikhaylov, A. A drone-based data management and optimization using metaheuristic algorithms and blockchain smart contracts in a secure fog environment. *Comput. Electr. Eng.* **2022**, *102*, 108234. [CrossRef]
8. Fernández-Caramés, T.M.; Blanco-Novoa, O.; Froiz-Míguez, I.; Fraga-Lamas, P. Towards an Autonomous Industry 4.0 Warehouse: A UAV and Blockchain-Based System for Inventory and Traceability Applications in Big Data-Driven Supply Chain Management. *Sensors* **2019**, *19*, 2394. [CrossRef]
9. Pan, N.; Chen, Q.; Liu, H.; Sun, Y.; An, Y.; Pan, D. Task planning of UAV stocktaking tray in complex industrial storage environment. *Comput. Integr. Manuf. Syst. CIMS* **2021**, *27*, 2940–2949. [CrossRef]
10. Chen, C.; Dong, D. Quantum intelligent mobile system. *Stud. Comput. Intell.* **2008**, *121*, 77–102. [CrossRef]
11. Udvaros, J.; Gubán, Á.; Gubán, M. Methods of artificial intelligence in economical and logistical education. In Proceedings of the eLearning and Software for Education Conference, Bucharest, Romania, 11–12 April 2019; pp. 414–421. [CrossRef]
12. Chen, W.; Meng, X.; Liu, J.; Guo, H.; Mao, B. Countering Large-Scale Drone Swarm Attack by Efficient Splitting. *IEEE Trans. Veh. Technol.* **2022**, *71*, 9967–9979. [CrossRef]
13. Wang, X.; Yang, S. Improved RRT algorithm path planning combined with artificial potential field algorithm. In Proceedings of the 11th International Workshop on Computer Science and Engineering (WCSE 2021), Shanghai, China, 19–21 June 2021; pp. 133–137. [CrossRef]
14. Xu, J.; Tian, Z.; He, W.; Huang, Y. A Fast Path Planning Algorithm Fusing PRM and P-Bi-RRT. In Proceedings of the 11th International Conference on Prognostics and System Health Management, PHM-Jinan, Jinan, China, 23 October 2020; Volume 9296702, pp. 503–508. [CrossRef]
15. Saeed, R.A.; Omri, M.; Abdel-Khalek, S.; Ali, E.S.; Alotaib, M.F. Optimal path planning for drones based on swarm intelligence algorithm. *Neural Comput. Appl.* **2022**, *34*, 10133–10155. [CrossRef]
16. Huang, S.-H.; Huang, Y.-H.; Blazquez, C.A.; Chen, C.-Y. Solving the vehicle routing problem with drone for delivery services using an ant colony optimization algorithm. *Adv. Eng. Inform.* **2022**, *51*, 101536. [CrossRef]
17. Albattah, W.; Javed, A.; Nawaz, M.; Masood, M.; Albahli, S. Artificial Intelligence-Based Drone System for Multiclass Plant Disease Detection Using an Improved Efficient Convolutional Neural Network. *Front. Plant Sci.* **2022**, *13*, 808380. [CrossRef]
18. Palossi, D.; Zimmerman, N.; Burrello, A.; Conti, F.; Muller, H.; Gambardella, L.M.; Benini, L.; Giusti, A.; Guzzi, J. Fully Onboard AI-Powered Human-Drone Pose Estimation on Ultralow-Power Autonomous Flying Nano-UAVs. *IEEE Internet Things J.* **2022**, *9*, 1913–1929. [CrossRef]
19. Samir, M.; Sharafeddine, S.; Assi, C.M.; Nguyen, T.M.; Ghrayeb, A. UAV Trajectory Planning for Data Collection from Time-Constrained IoT Devices. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 34–46. [CrossRef]
20. Jeong, S.; Simeone, O.; Kang, J. Mobile Edge Computing via a UAV-Mounted Cloudlet: Optimization of Bit Allocation and Path Planning. *IEEE Trans. Veh. Technol.* **2018**, *67*, 2049–2063. [CrossRef]
21. Wang, H.; Wang, J.; Ding, G.; Chen, J.; Gao, F.; Han, Z. Completion Time Minimization with Path Planning for Fixed-Wing UAV Communications. *IEEE Trans. Wirel. Commun.* **2019**, *18*, 3485–3499. [CrossRef]
22. Han, Y.; Chen, Q.; Pan, N.; Guo, X.; An, Y. Unmanned Aerial Vehicle 3D Trajectory Planning Based on Background of Complex Industrial Product Warehouse Inventory. *Sens. Mater.* **2022**, *34*, 3255–3269. [CrossRef]
23. Radácsi, L.; Gubán, M.; Szabó, L.; Udvaros, J. A Path Planning Model for Stock Inventory Using a Drone. *Mathematics* **2022**, *10*, 2899. [CrossRef]
24. Fülöp, M.T.; Gubán, M.; Gubán, Á.; Avornicului, M. Application Research of Soft Computing Based on Machine Learning Production Scheduling. *Processes* **2022**, *10*, 520. [CrossRef]
25. Gubán, M.; Gubán, Á. Production scheduling with genetic algorithm. *Adv. Logist. Syst. Theory Pract.* **2012**, *6*, 33–44.
26. Mraz, E.; Rodina, J.; Babinec, A. Using fiducial markers to improve localization of a drone. In Proceedings of the 2020 23rd IEEE International Symposium on Measurement and Control in Robotics 2020, ISMCR, Budapest, Hungary, 15 October 2020. [CrossRef]
27. Wawrla, L.; Maghazei, O.; Netland, T. *Applications of Drones in Warehouse Operations*; Whitepaper; Chair of Production and Operations Management Department of Management, Technology and Economics ETH Zurich: Zurich, Switzerland, 2019.
28. Gerwen, J.V.; Geebelen, K.; Wan, J.; Joseph, W.; Hoebeke, J.; De Poorter, E. Indoor Drone Positioning: Accuracy and Cost Trade-Off for Sensor Fusion. *IEEE Trans. Veh. Technol.* **2022**, *71*, 961–974. [CrossRef]
29. Chhikara, P.; Tekchandani, R.; Kumar, N.; Chamola, V.; Guizani, M. DCNN-GA: A Deep Neural Net Architecture for Navigation of UAV in Indoor Environment. *IEEE Internet Things J.* **2021**, *8*, 4448–4460. [CrossRef]

30. De Falco, A.; Narducci, F.; Petrosino, A. An UAV autonomous warehouse inventorying by deep learning. *Lect. Notes Comput. Sci.* **2019**, *11751*, 443–453. [CrossRef]
31. Nguyen, T.; Huynh, T.; Vu Ngoc, C.; Kieu, V.; Huang, S.C. Optimizing compliant gripper mechanism design by employing an effective bi-algorithm: Fuzzy logic and ANFIS. *Microsyst. Technol.* **2021**, *27*, 3389–3412. [CrossRef]
32. Wang, C.N.; Yang, F.C.; Nguyen, V.; Vo, N. CFD Analysis and Optimum Design for a Centrifugal Pump Using an Effectively Artificial Intelligent Algorithm. *Micromachines* **2022**, *13*, 1208. [CrossRef]
33. Yoon, B.; Kim, H.; Youn, G.; Rhee, J. 3D position estimation of drone and object based on QR code segmentation model for inventory management automation. In Proceedings of the 2021 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR), New York City, NY, USA, 25–27 October 2021; pp. 223–229. [CrossRef]
34. Kessler, R.; Melching, C.; Goehrs, R.; Gómez, J.M. Using camera-drones and artificial intelligence to automate warehouse inventory. In Proceedings of the AAAI 2021 Spring Symposium on Combining Machine Learning and Knowledge Engineering (AAAI-MAKE 2021), Stanford University, Palo Alto, CA, USA, 22–24 March 2021; Volume 2846.
35. Lu, J.; Zhao, L.; Tang, H. Three dimensional path planning on unmanned aerial vehicle based on radio frequency identification inventory management. *Comput. Integr. Manuf. Syst.* **2018**, *24*, 3129. [CrossRef]
36. Liu, H.; Chen, Q.; Pan, N.; Sun, Y.; An, Y.; Pan, D. UAV Stocktaking Task-Planning for Industrial Warehouses Based on the Improved Hybrid Differential Evolution Algorithm. *IEEE Trans. Ind. Inform.* **2022**, *18*, 582–591. [CrossRef]
37. Beul, M.; Droeschel, D.; Nieuwenhuisen, M.; Quenzel, J.; Houben, S.; Behnke, S. Fast Autonomous Flight in Warehouses for Inventory Applications. *IEEE Robot. Autom. Lett.* **2018**, *3*, 3121–3128. [CrossRef]
38. Tubis, A.A.; Ryczyński, J.; Żurek, A. Risk assessment for the use of drones in warehouse operations in the first phase of introducing the service to the market. *Sensors* **2021**, *21*, 6713. [CrossRef]
39. Geodis and Delta Drone Develop Autonomous Warehouse Solution. Available online: https://dronebelow.com/2018/04/12/geodis-and-delta-drone-develop-autonomous-warehouse-solution/ (accessed on 15 October 2022).
40. Gubán, M.; Kása, R.; Takács, D.; Avornicului, M. Trends of Using Artificial Intelligence. In Measuring Innovation Potential. *Manag. Prod. Eng. Rev.* **2019**, *10*, 3–15. [CrossRef]
41. Tusor, B.; Takáč, O.; Molnar, A.; Gubo, S.; Varkonyi-Koczy, A.R. Shape Recognition in Drone Images Using Simplified Fuzzy Indexing Tables. In Proceedings of the 2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI), Herlany, Slovakia, 23–25 January 2020; pp. 129–134. [CrossRef]
42. Gubo, S.; Kmet, T.; Molnar, A.; Takáč, O. A Multi-range Approach for Cultural Heritage Survey: A Case Study of a Medieval Church in Slovakia. In Proceedings of the IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI), Herlany, Slovakia, 23–25 January 2020; pp. 117–122. [CrossRef]
43. Tusor, B.; Bukor, J.; Végh, L.; Takáč, O. Domain Reduction Techniques for Sequential Fuzzy Indexing Tables—A Case Study. *Lect. Notes Netw. Syst.* **2020**, *101*, 179–190. [CrossRef]