

Article

A Path Planning Model for Stock Inventory Using a Drone

László Radácsi , Miklós Gubán , László Szabó * and József Udvaros

Faculty of Finance and Accountancy, Budapest Business School, 1149 Budapest, Hungary

* Correspondence: szabo.laszlo4@uni-bge.hu

Abstract: In this study, a model and solution are shown for controlling the inventory of a logistics warehouse in which neither satellite positioning nor IoT solutions can be used. Following a review of the literature on path planning, a model is put forward using a drone that can be moved in all directions and is suitable for imaging and transmission. The proposed model involves three steps. In the first step, a traversal path definition provides an optimal solution, which is pre-processing. This is in line with the structure and capabilities of the warehouse. In the second step, the pre-processed path determines the real-time movement of the drone during processing, including camera movements and image capture. The third step is post-processing, i.e., the processing of images for QR code identification, the interpretation of the QR code, and the examination of matches and discrepancies for inventory control. A key benefit for the users of this model is that the result can be achieved without any external orientation tools, relying solely on its own movement and the organization of a pre-planned route. The proposed model can be effective not only for inventory control, but also for exploring the structure of a warehouse shelving system and determining empty cells.

Keywords: drone; inventory management; GA model; route planning; warehouse**MSC:** 90B05

Citation: Radácsi, L.; Gubán, M.; Szabó, L.; Udvaros, J. A Path Planning Model for Stock Inventory Using a Drone. *Mathematics* **2022**, *10*, 2899. <https://doi.org/10.3390/math10162899>

Academic Editors: Zsolt Tibor Kosztán, Zoltán Kovács and Ripon Kumar Chakraborty

Received: 3 July 2022

Accepted: 10 August 2022

Published: 12 August 2022

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



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In large, lightly structured warehouses in logistics centers, especially those where different products from several companies are stored, it is often difficult to pinpoint the exact location of stored goods. This is primarily the case when storage is not carried out with the help of automatic forklifts, or when errors occur in the registered and actual location of the goods due to mistakes during the picking process. To make matters worse, it is very difficult or in some cases impossible to use GPS-based identifications in these warehouses. If Time of Flight (ToF) cameras are not available, the positioning of the automatic devices is not possible. In this study, a model capable of updating the stock-on-hand registry using a drone is introduced, and the proposed model applies to a specific warehouse.

Storage (loading and unloading) and picking, unit load formation, and labeling take place in the warehouse building. In the warehouse, a double shelving system is used. Between the double shelves, there is a wide corridor in which the drone can travel. A shelf is divided into rows or compartments (cells). Within one compartment there can be one, two, or three locations created. Each unit stored in these locations is identified with a QR code that contains all the important data that are relevant to our solution.

Driving down the aisle, the drone searches for the QR codes placed on the outer cartons, and as soon as the drone camera finds one, a photo is taken and it is sent to a processing device (in our study, it is a large capacity tablet). According to the proposed model, the drone knows its exact location in the warehouse, i.e., the 3D data inside the warehouse are defined. This and the QR code or codes sent (in the case of a multi-cell location) can tell the inventory management model exactly where the product is located, or if the location is empty. Further processing depends solely on the capabilities of the application, which is beyond the scope of this study.

The mid-range drone and its camera can take a good quality picture from the middle of the aisle and identify the stored goods in the picture based on the blue and red colors used on the shelves. This helps the drone to move forward, backward, up, and down in the middle of the aisle, with no right-to-left movement. At the end of the aisle, the dock is manually placed where the battery can be charged and replaced.

After this introduction, an optimal route is described that is optimized primarily for the operating time of the drone, i.e., the drone will continue checking the locations until its battery wears out.

Based on the literature review, we have identified a gap in the research, as we have not encountered an approach such as ours to inventory accounting; therefore, we can say that the drone inventory based on this paper can be considered as a development in the existing discussion in this field.

Our study is founded on the following research question: How can warehouse stocking be automated by drones? Within this question, we pose a sub-question: How can the time of stocking be optimized by drones? In this study, we aimed to develop an algorithm to mathematically answer these questions; thus, we have proposed a model for automated warehouse stocking.

2. Path Planning

Our analysis suggests that there are very different approaches to path planning for the problem we are considering. In general, these studies focus primarily on a specific problem. Furthermore, the solutions that are used also vary (e.g., [1]).

To begin with, we investigated several solution methods. According to the literature, studies have used the Swarm Intelligence approach to solve, for example, the collision avoidance problem of robots in 3D media such as water [2] using a velocity-matching method. The route planning used in this paper is based on the ant algorithm. Our investigations have shown that this solution does not provide sufficient convergence and that the motions in our problem are different from it, but several elements of the basic model can be incorporated into our model. A similar and similarly unsuccessful attempt based on swarm intelligence is presented in [3], where an evolutionary algorithm was also chosen as the suitable approach. Another approach that can be used concerns automatic parking systems. It essentially provides a similar baseline for the path selection in a planar domain to the vertical-plane path-search algorithm. Of course, the authors focus primarily on the narrow setting, i.e., they focus on maneuvering, and their path-finding algorithm focuses on finding a parking space, although this is solved using a less well-known RRT (Rapid-exploring Random Trees) algorithm. The chosen method for the model in [4] provides a useful solution in this direction. In this paper, the route planning of UAVs (Unmanned Aerial Vehicles) under terrain conditions is considered. The path is based on a polynomial model, a solution whereby only discontinuous linear motions need to be simulated. The advantage of their method is that it improves the initial population after construction using an ACO.

Several articles deal with logistics inventory control models and propose a specific solution. In their article, F. Benes et al. describe that, in the case of large outdoor warehouses, general identification methods are lengthy and inadequate. One way to easily and quickly determine the inventory is to deploy a UAV (unmanned aerial vehicle) for product identification. In this case, however, there is a problem in determining the location of the goods. A drone moves at a higher altitude, which can lead to a situation where we cannot accurately determine the location of the goods. This paper deals with the development of the definition of the correct flight level, which is suitable for distinguishing one of the identified elements at a distance of at least 2 m. The evaluation is based on an RSSI (received signal strength indicator) value. The experiment proved that the two objects can be distinguished even at the maximum reading distance of the selected passive UHF RFID tags [5]. This solution does not provide a suitable solution, since in this case the products

are placed in one flat, whereas we need to check the inventory of a warehouse with a shelf system (3D).

Indoor drone or unmanned aerial vehicle (UAV) operations in automated or pilot-controlled drone use cases are addressed by Kurt Geebelen et al. Automated indoor flights have stricter requirements for stability and localization accuracy compared to classic outdoor use cases, which rely primarily on (RTK) GNSS for localization. In this paper, the effect of multiple sensors on 3D indoor-position accuracy is investigated using the OASE flexible sensor fusion platform. This evaluation is based on real drone flights in an industrial laboratory, with mm-accurate ground truth measurements provided by motion capture cameras, which enable the sensors to be evaluated based on their deviation from ground reality in 2D and 3D. The sensors considered in the research are: IMU, the sonar, SLAM camera, ArUco markers, and Ultra-Wideband (UWB). The article shows that with this setup, the achievable 2D (3D) indoor localization error varies from 4.4 cm to 21 cm depending on the sensor set selected. They also include cost/accuracy tradeoffs to indicate the relative importance of different sensor combinations depending on the (engineering) budget and use case. These laboratory results were validated in a Proof-of-Concept deployment of an inventory-scanning drone with more than 10 flight hours in a 65,000 m² warehouse. By combining the laboratory results and real-world deployment experience, the different subsets of the sensors represent a minimum viable solution 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 swarm of weight- and cost-constrained nano drones [6]. Our solution also eliminates these flight inaccuracies, as the drone takes a picture from the approaching position, which is processed by software (which extracts the QR code from it). Therefore, the inaccuracies recommended in the article are negligible in our case.

As part of their research project, A. Rhiat, L. Chalal and A. Saadane developed a prototype named “Smart Shelf” that simulates a smart warehouse where mobile robots with grips managed by the ROS “Robotic Operating System” can autonomously navigate through inverse kinematics and different obstacles between other robots; on the other hand, RFID and iBeacon technology have a Smart Shelf to manage stocks. All items on the shelf are identified by RFID tags. The connection of the robots must be applicable to the various predefined or non-obstacles to optimize their search and items they encounter and to accessing the local network through a predefined map in the database. In addition, the embedding of Bin Packing Optimization techniques helps to improve the utilization of static volumes. Optimization algorithms can take into account some robotic constraints, including accessibility, improving the quality of placements and minimizing damaged goods. Their project aims to minimize human intervention and gain time [7]. We consider the solution very useful; it can also be used in our case. However, a problem is caused by the fact that we are inspecting the stock during the storage process, while material handling is taking place. Therefore, we have to discard this solution as well, since it is not possible to use the Smart Shelf during the material-handling process.

Haishi Liu et al. show that tobacco companies must regularly take inventory of finished products as well as raw and auxiliary materials, and drones with radio frequency identification (RFID) readers are becoming a major application trend for inventory applications. Under the condition of ensuring the accuracy of the inventory, this paper considers the limitations of the drone’s physical performance, the limitations of the RFID reader, etc., and presents the power of the drone with respect to modeling and a task-planning model for the UAV. An inventory library equipped with an RFID reader is recommended. Thus, considering the problem whereby the greedy strategy in the traditional differential evolution (DE) algorithm causes the loss of location information preserved by other individuals, we propose a hybrid DE algorithm based on lion swarm optimization. Finally, the proposed algorithm was verified by environmental modeling based on data from the tobacco industry warehouse [8]. This solution is already similar to our model, but our proposed Generation Algorithm (OGA) simplifies the solution.

Our study also concerns the determination of optimal routes. The location search strategies used during picking can also provide a good basis, as the goal is to work as quickly and accurately as possible. Based on this, two basic strategies are distinguished: in the case of the strategy regarding the increasing horizontal location coordinates, the order of visiting each storage location corresponds to the increase in the horizontal coordinates of the storage sites. The zoning strategy is a variation of the strategy of increasing the horizontal location coordinates, in which the scaffolding field is first divided into (even-numbered) superimposed zones, and storage locations within each zone are accessed in ascending order of the horizontal location coordinates [9]. However, regarding warehouse robotization, in many cases, besides route planning, the other task is to balance the size of the robot fleet and the load stabilizing between the robots, which is not the case of this study [10].

Moving to the next aisle is a manual movement, as the aisles may not be processed in a sequential way, there may be a closed aisle, or the next aisle might be in another part of the building. This adds another aim to our proposed model of enabling the drone to recognize its own position.

These types of solutions are already used in several places, mainly for large multinational companies. There are some excellent solutions such as the Eyesee system developed by the Hardis Group, a comprehensive drone inventory solution that includes a drone capable of flying unmanned and equipped with a system for automatically capturing and identifying barcode data and handling the collected data via Amazon Web Services [11].

A similar problem is analyzed in [12], involving the target tracking of a drone moving in a defined closed area, but this goes beyond the scope of our study, as their goal is to recognize the target and choose its speed; however, it is still of relevance. The article chooses the Fuzzy solution, which is a viable option for this study as well, since in many cases the position of the QR code within the given location is not clear and the camera unfortunately does not see it in all cases. For this reason, fuzzy control after the basic and deterministic movement should be incorporated into the model. This can also be applied to differences due to columns. The authors of [13] solve the positioning of a drone in a GPS-free environment using a ToF camera used in robotics and self-guidance. However, this approach demands considerable investment into hardware as well. The camera is installed on the ceiling, and this monitors the positioning of the drone in the x, y direction and changes in height. Important technical solutions are presented to prevent interference from rotors. With the use of Gaussian function filtering, it was feasible to accurately analyze the situation in 3D. The article also provides an exact algorithm as a solution. Another methodology worth considering for closed-area drone control is presented in [14]. The authors use a voxel model and perform two types of route calculations: one for the shortest route and the other for the cheapest route.

Their problem is mainly the use of obstacle-avoidance image analysis, for which they use the distance transformation method for an abnormal image laid down by A. Rosenfeld and J. L. Pfaltz in *Distance Functions on Digital Pictures*, from the journal *Pattern Recognition* [15]. The obstacles, in the form of shelves, are assumed to be fixed, which is problematic because this is not always the case. Their approach keeps the drone safe from the obstacle within an effective range. Another article suggests a search for known trajectory options [16]. In their proposal, a usable genetic algorithm is used, which would be very useful in choosing the optimal crawl route for our study as well. The initial population is selected by a randomly generated greedy algorithm. An interesting solution is a low-complexity, machine-learning-based algorithm for optimizing the location of DBSs (drone base stations) [17] based on minimizing the collective wireless reception signal strength experienced by active terminals. The proposed algorithm reduces propagation loss in the system and provides a lower bit error rate compared to the Euclidean cost comparison. The main result of this study is the creation of a model that can provide input for a genetic Algorithm and can even be effective as a precise tool. This is pre-operation processing, as all processing can start with the knowledge of the optimum. The second result is a clear position-definition that is in-service and real-time.

3. Materials and Methods

The first step in creating a model is to clarify the notations and their content; therefore, a complete parameter exploration was carried out. The data set was defined with the help of practitioners. The basic data collected were grouped according to their roles in the model. Then, after selecting the parameters needed, it was important to capture their dimensions. In the second step, we created a parametric model of the warehouse and the factory parameters of the drone we wanted to use, improved with the data we had found. We created a manageable simplified model of motion and velocity (the simplification was within reason). The next step was to work out a practical positioning strategy. Then, we created a mathematical model of the operation. After that, the necessary items and environment for the optimization were selected.

During the testing of the GA method, we encountered some inconsistencies, which caused the model to undergo several manual tunings (the main reasons were the discrepancy between the factory and real drone parameters and the not necessarily smooth motion, which could have been due to several reasons, e.g., temperature and stock saturation problems). These inconsistencies, as they required relatively small modifications, were easily eliminated.

3.1. Known Data

In this section, we provide the mathematical model of the problem and the associated known and unknown data. The known data include the most important parameters of the warehouse and the drone: specifically, they were provided by the warehousing company, and the drone parameters are given in the technical specifications. Tables 1 and 2 summarize the data used for the model and provide the notations. We will use each index consistently for the same data, so we will also provide a table of indexes.

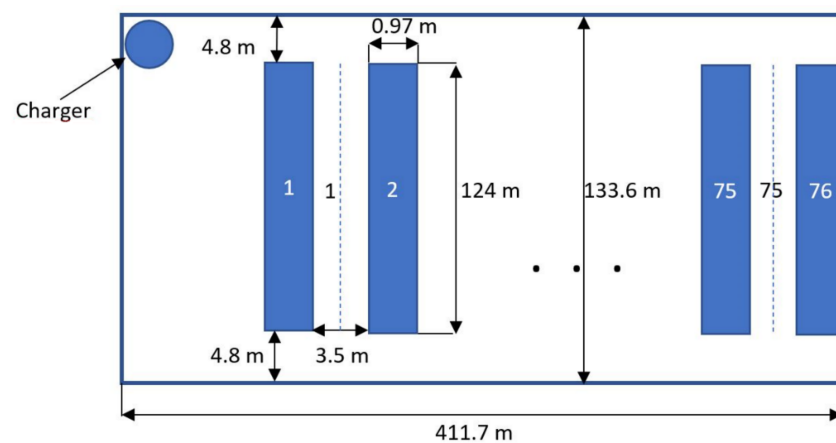
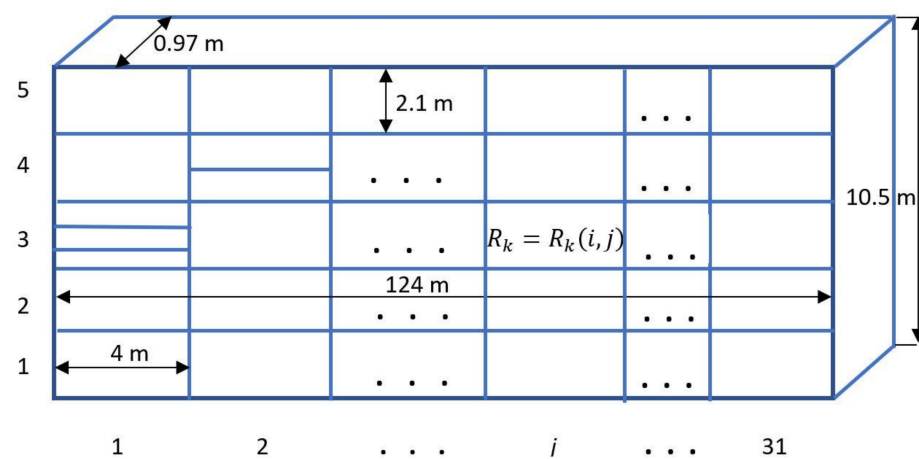
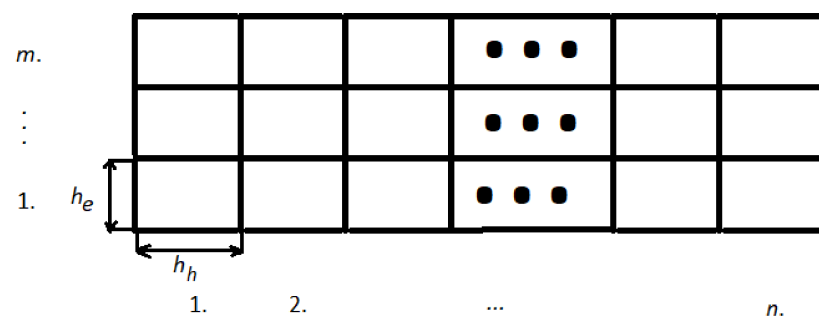
Table 1. Summary and notation of known data.

$\mathbf{w} = \mathbf{w}[w_x, w_y]$	The size of the warehouse
h_e	The vertical distance between the storage shelves
h_h	Horizontal distance between storage shelves
m	The number of storage shelves
h_w	The width of the corridor
h_c	The width of the road
n	The number of compartments within the rows
r	The number of rows of shelves
$\mathbf{D} = \mathbf{D}[d_x, d_y, d_z]$	The position of the dock. $d_z = 0$
$\mathbf{P}_l = \mathbf{P}_l[p_{lx}, p_{ly}, 0]$	The position of shelf $\mathbf{P}_l$ is indicated by the red circle in Figure 2. If l is odd, it denotes the left shelf row of the aisle; if l is even, it denotes the right shelf row.
$R_k = R_k(i, j)$	Shows the compartment in the i -th row and j -th column of the k -th shelf ($k = 1; 2$)
$\mathbf{C}_k = \mathbf{C}_k[c_{ix}, c_{iy}, c_{iz}]$	The starting position of the middle guide path between the two shelves, e.g., $c_{kx} = \frac{p_{2l+1,x} + p_{2l+2,x}}{2}$, etc., and ($l = 0, \dots, r-1$). In Figure 2, it is indicated by a green square. $c_{kz} = \frac{h_e}{2}$, where the z coordinate is equal to half the height of the shelf.
$\bar{\mathbf{C}}_k = \bar{\mathbf{C}}_k[\bar{c}_{ix}, \bar{c}_{iy}, \bar{c}_{iz}]$	The end position of the middle guide path between the two shelves. $\bar{c}_{kx} = \frac{p_{2l+1,x} + p_{2l+2,x}}{2}$, etc., and ($l = 0, \dots, r-1$). In Figure 2, it is indicated by a green square. $\bar{c}_{kz} = \frac{h_e}{2}$. Here, the z coordinate is equal to half the height of the shelf.
T_w	At full charge, the operating time of the drone. During this time, a person must get from the charger to the compartments and back
T_{PH}	Time for taking a picture
V_h	Horizontal forward speed
V_a	Speed of ascent
V_d	The rate of descent is $V_a \approx 0.99 V_d$; therefore, the two are considered to be equal.
V_r	Rotation speed (degrees)
Δt	Indicates the operating time that must be left on arrival at the charger in case of intermediate charging.
T_{ch}	Charging time

Table 2. Table of indexes.

k, k_1	The number of the shelving system
i, i_1	Indicate the rows of the shelving system
j, j_1	Indicate the columns of the shelving system (as an example, see the $R_k = R_k(i, j)$ in the previous table)
v	Indicates the subindex of the consecutive compartments
u	The running index of the charges

The schematic structure of the shelving system is shown in Figures 1–4.

**Figure 1.** The floor plan of the warehouse with the dimensions. (Source: self-edited figure).**Figure 2.** Shelf system details. (Source: self-edited figure).**Figure 3.** Structure of a shelf. (Source: self-edited figure).

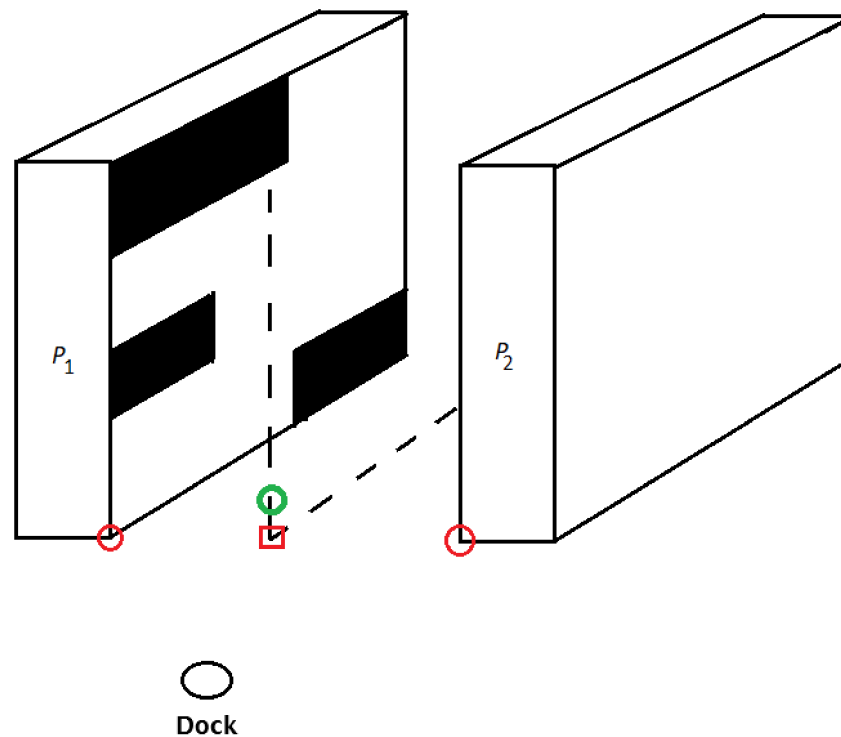


Figure 4. The shelving system to be tested. Black shelves indicate compartments that have already been checked. The green circle is the start position of the examination, the red square is the center of the corridor (reference point). The red circles are reference points. (Source: self-edited figure).

3.2. Variable Data

Search for the

$$S = S((k_1 i_1 j_1), (k_2 i_2 j_2), \dots, (k_{2mn} i_{2mn} j_{2mn})) \quad (1)$$

series of compartments, where

$$(k_{l_1} i_{l_1} j_{l_1}) \neq (k_{l_2} i_{l_2} j_{l_2}) \text{ ha } l_1 \neq l_2. \quad (2)$$

The drone passes over all the compartments of the two shelving systems during operation. The series of compartments S describes the order of these compartments where the first parameter is the shelf system number (1 or 2) and the other two are the number of shelves (rows) in the shelf system and the number of compartments (columns) within it.

The drone needs to be charged from time to time so it may not be possible to process the two shelf systems with one charge (especially when the batteries are not fully charged from the start). For this reason, the series of compartments S must be divided into parts that can be processed with one charge, i.e., the drone contains compartments processed during operation. (Table 3 shows summary and notation of variable data). Thus, the processing time consists of three parts:

- The drone travels to the first compartment to be processed;
- It processes the compartments;
- It returns to the charger.

Table 3. Summary and notation of variable data.

$\mathbf{x} = [x_{k,i,j}]$	The number of processing sequence (as the drone moves between each compartment)
$T_f = T_f(S)$	The execution time for the S -compartment series visit
$t_P = t_P(\acute{S})$	The dockless execution time of the $\acute{S}$ series
$t_{T,R_k(i_p,j_p)}$	The time from the dock to the starting point (i_p, j_p)
$t_{R_k(i_q,j_q),T}$	The travel time from (i_q, j_q) to the dock
$t_{R_{k_1}(i_1,j_1),R_{k_2}(i_2,j_2)}$	The travel time from one compartment in one row to another compartment in another row
$T_{R_{k_1}(i_1,j_1),R_{k_2}(i_2,j_2)}(C)$	Time from one row compartment to the other row compartment with a front-of-row change
$T_{R_{k_1}(i_1,j_1),R_{k_2}(i_2,j_2)}(\bar{C})$	Time from one row compartment to another row compartment with end of row change
p	Number of charges—1

The following correlations illustrate the abovementioned 3 points. The drone goes to the charger when it arrives at 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 predefined level.

In addition, it must be fulfilled that the sum of each subchain must be equal to the entire chain, since each compartment must be examined in S order.

Mark the series of compartment tests carried out simultaneously in a corridor

$$\check{S} = \check{S}((k_p i_p j_p), \dots, (k_q i_q j_q)) \subset S \quad (q \geq p) \quad q, p \in \mathbb{N}^+ \quad (3)$$

if the shelving system is to be changed, it will change either in the direction of $\bar{C}$ or in the direction of C .

Let $\check{S}$ denote the corridor change in G , i.e.,

$$G = \begin{cases} (k_p, k_{p+1}, 0), & \text{if it changes in direction } C \\ (k_p, k_{p+1}, -1), & \text{if it changes in direction } \bar{C} \end{cases} \quad (4)$$

Let

$$\acute{S} = \acute{S}(\check{S}_1, G_1, \check{S}_2, \dots, G_{f-1}, \check{S}_f) \quad (5)$$

then, the execution time of the $\acute{S}$ series is from the charger to the charger

$$T_f(\acute{S}) = t_{T,R_k(i_p,j_p)} + t_P(\acute{S}) + t_{R_k(i_q,j_q),T} \quad (6)$$

if the following is met

$$0 \leq T_w - T_f(\acute{S}) \leq \Delta t \quad (7)$$

Let

$$S = S((k_1 i_1 j_1), (k_2 i_2 j_2), \dots, (k_{2mn} i_{2mn} j_{2mn})) = \sum_{u=1}^r \acute{S}_u \quad (8)$$

where every $\acute{S}_u$

$$0 \leq T_w - T_f(\acute{S}_u) \leq \Delta t \quad (u = 1, \dots, r-1). \quad (9)$$

The last one no longer needs to meet this condition.

The final report must also include a flight to the charger. For this reason, S must be supplemented with the **D** drone charger in the appropriate places. Its location will be determined by a later algorithm (Target Function Generation Algorithm). The function

$T_f(\dot{S}_u)$ represents the operating time of the drone from a charger to the next charger. The sum of these yields the total operating time.

Then the execution sequence is as follows:

$$S_v = S_v(\mathbf{D}, \dot{S}_1, \mathbf{D}, \dots, \dot{S}_r, \mathbf{D}). \quad (10)$$

Then the goal is

$$T = \sum_{u=1}^r T_f(\dot{S}_u) \rightarrow \min! \quad (11)$$

If $T_f(S_v)$ is minimal, then S_v is the optimal route

3.3. The Speed of the Drone

The speed of the drone is determined by knowing the speed of horizontal travel and the speed of vertical travel by projecting it on a diagonal path, as shown in Figure 5. The red arrow indicates velocity as a vector and its magnitude is the magnitude of the current velocity.

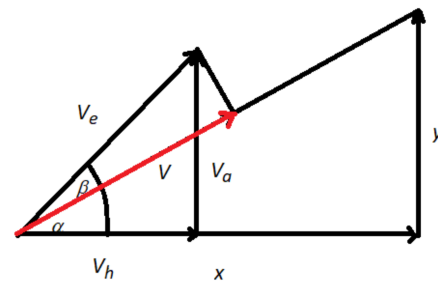


Figure 5. Drone speed (source: self-edited figure).

Ascension:

$$V_e = \sqrt{v_h^2 + v_a^2} \quad (12)$$

$$\alpha = \arctan \frac{y}{x}, \beta = \arctan \frac{v_a}{v_h} \quad (13)$$

Descension:

$$V_e = \sqrt{v_h^2 + v_d^2} \quad (14)$$

$$\beta = \arctan \frac{v_d}{v_h} \quad (15)$$

It follows that the ascension speed of the drone is:

$$V_1(x, y) = V_e \cdot \cos(\beta - \alpha) = V_e \cdot \cos\left(\arctan \frac{v_a}{v_h} - \arctan \frac{y}{x}\right) \quad (16)$$

It follows that the descension speed of the drone is:

$$V_2(x, y) = V_e \cdot \cos\left(\arctan \frac{v_d}{v_h} - \arctan \frac{y}{x}\right) \quad (17)$$

Let i_v, j_v denote the indices of the compartment corresponding to the location of the drone, and i_{v+1}, j_{v+1} denote the indices of the next compartment to be examined by the drone.

If

$$x = \text{abs}(j_v - j_{v+1}) \cdot h_h \quad (18)$$

$$y = \text{abs}(i_v - i_{v+1}) \cdot h_e. \quad (19)$$

If $x = 0$, then

If $i_v - i_{v+1} > 0$

$$V_3(x, y) = V_d \quad (20)$$

If $i_v - i_{v+1} < 0$

$$V_4(x, y) = V_a \quad (21)$$

then it follows that the speed of the drone is:

$$V = V(x, y) = \begin{cases} j_v - j_{v+1} \geq 0 \text{ és } x \neq 0 \text{ akkor } V_2(x, y) \\ j_v - j_{v+1} < 0 \text{ és } x \neq 0 \text{ akkor } V_1(x, y) \\ j_v - j_{v+1} \geq 0 \text{ és } x = 0 \text{ akkor } V_3(x, y) \\ j_v - j_{v+1} < 0 \text{ és } x = 0 \text{ akkor } V_4(x, y) \end{cases} \quad (22)$$

Thus, substituting $(i_v - i_{v+1}) \cdot h_h$ for x , and substituting $(j_v - j_{v+1}) \cdot h_e$ for y , we obtain the following

$$\begin{aligned} V_1 &= V_1((j_v - j_{v+1}) \cdot h_e, (i_v - i_{v+1}) \cdot h_h) = \\ &= V_e \cdot \cos\left(\arctan\left(\frac{v_a}{v_h}\right) - \arctan\left(\frac{\text{abs}(i_v - i_{v+1}) \cdot h_e}{\text{abs}(j_v - j_{v+1}) \cdot h_h}\right)\right) \end{aligned} \quad (23)$$

$$\begin{aligned} V_2 &= V_2((i_v - i_{v+1}) \cdot h_h, (j_v - j_{v+1}) \cdot h_e) = \\ &= V_d \cdot \cos\left(\arctan\left(\frac{v_d}{v_h}\right) - \arctan\left(\frac{\text{abs}(i_v - i_{v+1}) \cdot h_e}{\text{abs}(j_v - j_{v+1}) \cdot h_h}\right)\right) \end{aligned} \quad (24)$$

This depends on the horizontal and vertical travel distances.

3.4. The Length and Time of Each Route

The drone begins all processing by centering the two shelving systems, as shown in the green circle's position in Figure 6. This point denotes the center of the bottom row (which can be adjusted if necessary). The drone can take the best shot from the center of the compartments. The steps of processing include:

- The drone travels to the first compartment to be processed;
- It processes compartments;
- It travels back to the charger.

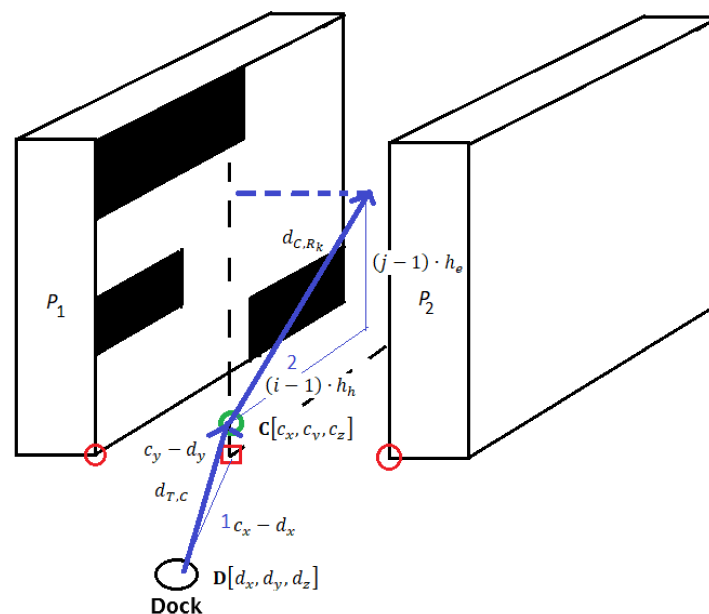


Figure 6. Route and guides. The green circle is the start position of the examination; the blue line is the route of drone; the red square is the center of the corridor (reference point). The red circles are referece points. (source: self-edited figure).

3.4.1. The Journey and Time from the Dock to the Starting Point

It consists of three parts (Figure 6):

- Step 1. the drone travels from the charger to the starting point between the shelves (as indicated by the green circle);
- Step 2. the drone travels to the first compartment to be processed;
- Step 3. its camera is turned in the correct direction.

Step 1. Time to reach the starting position of the shelves

Distance of compartment $R_k = R_k(i, j)$ from charger:

$$d_{T,R_k(i,j)} = d_{T,C} + d_{C,R_k}. \quad (25)$$

$R_k = R_k(i, j)$ compartment access time from charger:

$$t_{T,R_k(i,j)} = t_{T,C} + t_{C,R_k}, \quad (26)$$

where $d_{T,C}$ is the path between the docking station and the starting position of the middle guide path between the two shelves to the green square.

$$d_{T,C_k} = \sqrt{\left(\frac{h_e}{2}\right)^2 + (c_{kx} - d_x)^2 + (c_{ky} - d_y)^2} \quad (27)$$

$t_{T,C}$ is the time required to reach the starting position of the middle guide path between the Dock and the two shelves:

$$t_{T,C_k} = \frac{h_e}{2V_a} + \frac{\sqrt{(c_{kx} - d_x)^2 + (c_{ky} - d_y)^2}}{V_h}. \quad (28)$$

Step 2. Distance and time from the starting position of the shelves to the starting compartment.

Distance and time from the center to the starting compartment:

$$d_{C_k,R_k} = \sqrt{((j-1) \cdot h_h)^2 + ((i-1) \cdot h_e)^2} \quad (29)$$

$$t_{C_k,R_k} = \frac{d_{C_k,R_k}}{V((j-1) \cdot h_h, (i-1) \cdot h_e)} + \frac{90}{V_r} + T_{PH} \quad (30)$$

Step 3. The camera

Where $\frac{90}{V_r}$ is the camera rotation time.

The way to the starting position:

$$\begin{aligned} t_{T,R_k(i,j)} &= t_{T,C_k} + t_{C_k,R_k} = \\ &= \frac{h_e}{2V_a} + \frac{\sqrt{(c_{kx} - d_x)^2 + (c_{ky} - d_y)^2}}{V_h} + \frac{\sqrt{((j-1) \cdot h_h)^2 + ((i-1) \cdot h_e)^2}}{V((j-1) \cdot h_h, (i-1) \cdot h_e)} + \frac{90}{V_r} + T_{PH} \end{aligned} \quad (31)$$

3.4.2. The Time to Travel to the Dock from a Given Point Can Be Determined Similarly

This is the reverse of the previous since

- Step 1—first, rotate the camera to the home position.
- Step 2—go to the starting point of the shelving system.
- Step 3—go to the charger.

Summarized as follows:

$$t_{R_k(i,j),T} = t_{R_k,c} + t_{C_k,T} = \frac{90}{V_r} + \frac{\sqrt{((j-1) \cdot h_h)^2 + ((i-1) \cdot h_e)^2}}{V((j-1) \cdot h_h, (i-1) \cdot h_e)} + \frac{h_e}{2V_d} + \frac{\sqrt{(c_{kx} - d_x)^2 + (c_{ky} - d_y)^2}}{V_h} \quad (32)$$

3.4.3. The Length and Time of a Point-To-Point Route

The path and time between the next two compartments are determined as shown in Figure 7. Due to the middle plane, it is sufficient to determine the distance between the two compartments by creating a Pythagorean theorem. It is enough to calculate the speed for the time and by dividing the distance we obtain the time. In addition, the camera can rotate if necessary (180°). This can be determined by subtracting the position of the shelf system of the two compartments and then taking the absolute value. This is 0 if the 2 cells are in the same shelving system and 1 if the two compartments are in a separate shelving system. If the rotation time is multiplied by this value, it will either rotate or not as per our need. At the end, the time of photography is added.

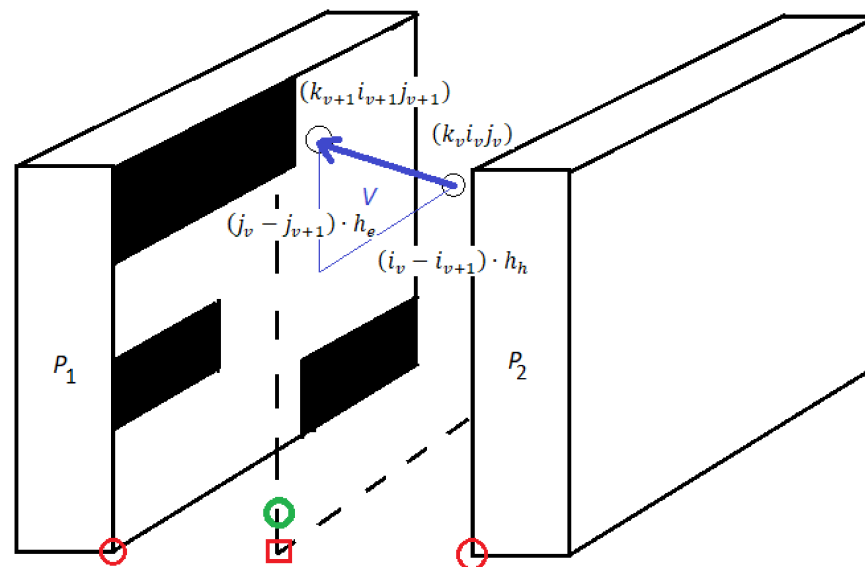


Figure 7. Compartment to compartment. The green circle is the start position of the examination; the blue line is the route of drone; the red square is the center of the corridor (reference point). The red circles are reference points. (source: self-edited figure).

Let us take $(k_v i_v j_v), (k_{v+1} i_{v+1} j_{v+1})$ as the two points. Their distance is:

$$d((k_v i_v j_v), (k_{v+1} i_{v+1} j_{v+1})) = \sqrt{((j_v - j_{v+1}) \cdot h_h)^2 + ((i_v - i_{v+1}) \cdot h_e)^2} \quad (33)$$

Time spent on this route by the drone:

$$t_{P_1}((k_v i_v j_v), (k_{v+1} i_{v+1} j_{v+1})) = \frac{\sqrt{((j_v - j_{v+1}) \cdot h_h)^2 + ((i_v - i_{v+1}) \cdot h_e)^2}}{V((j_v - j_{v+1}) \cdot h_h, (i_v - i_{v+1}) \cdot h_e)} + \frac{abs(k_v - k_{v+1}) \cdot 180}{V_r} + T_{PH} \quad (34)$$

where

$$\frac{abs(k_1 - k_2) \cdot 180}{V_r} \quad (35)$$

is the rotation of the camera.

3.4.4. From One Compartment to Another Compartment

When considering change from one compartment in one row to another compartment in another row, we used the following steps (See Figure 8):

- Step 1—first, the drone moves to the starting point of the shelving system;
- Step 2—then, it moves to the starting point of the next shelving system;
- Step 3—then, it moves to the starting compartment of the queue.

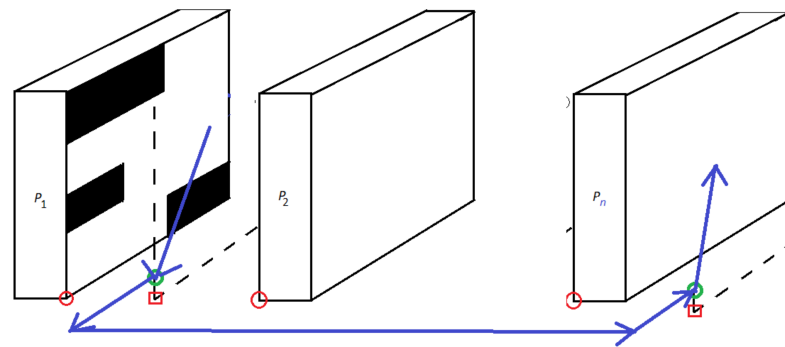


Figure 8. From compartment to compartment to another row. The green circle is the start position of the examination; the blue line is the route of drone; the red square is the center of the corridor (reference point). The red circles are referece points.

The situation is similar if the change is made at the other end of the shelving system. Let

$$a(j) = \begin{cases} j-1, & \text{if } C \text{ is the point under test} \\ n-j, & \text{if } \bar{C} \text{ is the point under test} \end{cases} \quad (36)$$

$$\begin{aligned} T_{R_{k_1}(i_1, j_1), R_{k_2}(i_2, j_2)}(C) &= t_{R_{k_1}, C_{k_1}} + t_{C_{k_1}, C_{k_2}} + t_{C_{k_1}, R_{k_1}} = \\ &= \frac{d_{R_{k_1}, C_{k_1}}}{V(a(j_1) \cdot h_h, (i_1-1) \cdot h_e)} + \frac{90}{V_r} + \frac{\frac{h_c}{2} + \text{abs}(k_1 - k_2) \cdot (h_w + h_d) + \frac{h_c}{2}}{v_h} + \\ &\quad + \frac{d_{C_{k_2}, R_{k_2}}}{V(a(j_1) \cdot h_h, (i_2-1) \cdot h_e)} = \\ &= \frac{d_{R_{k_1}, C_{k_1}}}{V(a(j_1) \cdot h_h, (i_1-1) \cdot h_e)} + \frac{h_c + \text{abs}(k_1 - k_2) \cdot (h_w + h_d)}{v_h} + \\ &\quad + \frac{d_{C_{k_2}, R_{k_2}}}{V(a(j_2) \cdot h_h, (i_2-1) \cdot h_e)} + \frac{90}{V_r} + T_{PH} \end{aligned} \quad (37)$$

Similar to the optimal path, calculate the

$$T_{R_{k_1}(i_1, j_1), R_{k_2}(i_2, j_2)}(\bar{C}) = t_{R_{k_1}, \bar{C}_{k_1}} + t_{\bar{C}_{k_1}, \bar{C}_{k_2}} + t_{\bar{C}_{k_1}, R_{k_1}} \quad (38)$$

Let

$$t_{R_{k_1}(i_1, j_1), R_{k_2}(i_2, j_2)} = \min(T_{R_{k_1}(i_1, j_1), R_{k_2}(i_2, j_2)}(C), T_{R_{k_1}(i_1, j_1), R_{k_2}(i_2, j_2)}(\bar{C})) \quad (39)$$

Taking into account that there may be two consecutive compartments to being examined within a row or between different rows, the function t_P is composed of two parts, of which only the current one is always included. If there are two opposing shelving systems, then if $k_{v+1} = k_v + 1$ and $\text{mod}(k_v, 2) = 1$, or $k_v = k_{v+1} + 1$ and $\text{mod}(k_v, 2) = 0$, or $k_{v+1} = k_v$, then

$$t_P((k_v i_v j_v), (k_{v+1} i_{v+1} j_{v+1})) = t_{P_1}((k_v i_v j_v), (k_{v+1} i_{v+1} j_{v+1})) \quad (40)$$

otherwise

$$t_P((k_v i_v j_v), (k_{v+1} i_{v+1} j_{v+1})) = t_{R_{k_v}(i_v, j_v), R_{k_{v+1}}(i_{v+1}, j_{v+1})} \quad (41)$$

Execution time of the $\dot{S}_u$ sub-sequence (without docking):

$$t_P(\dot{S}_u) = \sum_{v=p}^{q-1} t_P((k_v i_v j_v), (k_{v+1} i_{v+1} j_{v+1})). \quad (42)$$

$$T_f(\dot{S}_u) = t_{T, R_k(i, j)} + t_P(\dot{S}_u) + t_{R_k(i, j), T} \quad (43)$$

3.5. The Constraints

Based on the above, the sum of the compartment variables is exactly the sum of the number of cells from 0 to the number of cells -1 .

Two conditions must be provided in the model's conditional framework:

1. Each number of compartments must be different;
2. The row numbers must take all values starting from 0 up to the total number of cells (i.e., the row numbers must increase one by one);

This can be specified by two group conditions. The value of the variable cells must not be negative (since it is an entry sequence); therefore,

$$0 \leq x_{kij}. \quad (44)$$

The value of x_{kij} can be clearly bounded from above:

$$x_{kij} \leq r \cdot m \cdot n. \quad (45)$$

Each number must be different if you are looking at different compartments:

$$k, k_1 \in [1, \dots, r], i, i_1 \in [1, \dots, m], j, j_1 \in [1, \dots, n]. \quad (46)$$

If (i.e., two compartments are different), then

$$\begin{aligned} 1000k_1 + 100i_1 + j_1 &\neq 1000k_2 + 100i_2 + j_2, \\ x_{k_1, i_1, j_1} &\neq x_{k_2, i_2, j_2}. \end{aligned} \quad (47)$$

In summary, the mathematical model

$$\begin{aligned} k, k_1 &\in [1, \dots, r], i, i_1 \in [1, \dots, m], j, j_1 \in [1, \dots, n] \\ 0 &\leq x_{kij} \leq r \cdot m \cdot n \\ x_{kij} &\neq x_{k_2, i_2, j_2} \text{ if } 1000k + 100i + j \neq 1000k_1 + 100i_1 + j_1, \\ T &= \sum_{u=1}^p T_f(\dot{S}_u) \rightarrow \min! \end{aligned} \quad (48)$$

4. The Solution Method

The computational model can be further simplified by

$$0 \leq x_{kij} \leq r \cdot m \cdot n \quad (49)$$

The constraint can be replaced by the

$$0 \leq x_{kij} \leq mtp \cdot r \cdot m \cdot n \quad (50)$$

(e.g., may be $mtp = 100$), and use the following routine as the nonlinear constraint. This is important for the genetic algorithm. Then, it is sufficient that all values are different.

4.1. The Objective Generation Algorithm (OGA)

Step 1.

Construct a random set of

$$S = S((k_1 i_1 j_1), (k_2 i_2 j_2), \dots, (k_{2mn} i_{2mn} j_{2mn})) \quad (51)$$

compartments where

$$(k_{l_1} i_{l_1} j_{l_1}) \neq (k_{l_2} i_{l_2} j_{l_2}), \text{ if } l_1 \neq l_2. \quad (52)$$

Each compartment should be tested once and only once.

Step 2.

Let $u = 0$, $S_v = \emptyset$,

and $g = (k_1 i_1 j_1)$ the first item in the series (if any).

Step 3.

If there are no more items in the series, continue from Step 5.

If there are, then

$$\begin{aligned} u &:= u + 1 \\ \dot{S}_u &= \emptyset \end{aligned} \quad (53)$$

Step 4.

Add element g of the series to the subsequence.

$$\dot{S}_u := \dot{S}_u + \{g\} \quad (54)$$

Calculate the value of

$$T_f(\dot{S}_u) \quad (55)$$

If

$$T_w - T_f(\dot{S}_u) > \Delta t \quad (56)$$

take the next element g and continue from Step 4.

If

$$T_w - T_f(\dot{S}_u) \leq 0 \quad (57)$$

i.e., the drone would not return with the addition of the new element, then

$$\begin{aligned} \dot{S}_u &:= \dot{S}_u - \{g\} \\ T &:= T + T_f(\dot{S}_u) \\ S_v &:= S_v + \{\mathbf{D}, \dot{S}_u\} \end{aligned} \quad (58)$$

Then proceed to Step 3.

If

$$0 \leq T_w - T_f(\dot{S}_u) \leq \Delta t \quad (59)$$

Then take the next item g and continue from Step 3.

$$\begin{aligned} T &:= T + T_f(\dot{S}_u) \\ S_v &:= S_v + \{\mathbf{D}, \dot{S}_u\} \end{aligned} \quad (60)$$

Step 5.

Let

$$S_v := S_v + \{\mathbf{D}\} \quad (61)$$

At this point, the task list was completed with the charges, and the total processing time T was obtained.

4.2. Determining the Optimal Route

We used an evolutionary algorithm for the solution.

Step 1. Set the initial parameters (population number, number of steps, etc.). The formation of the initial population, i.e., each chromosome, will be one

$$S = S((k_1 i_1 j_1), (k_2 i_2 j_2), \dots, (k_{2mn} i_{2mn} j_{2mn})) \quad (62)$$

series.

Step 2.

Run the algorithm by determining the fitness value for each individual in the population using the Target Function Generation Algorithm (OGA). This should also be built into factory algorithms.

Step 3.

After stopping the algorithm, we evaluate the result and program the obtained optimal solution (S_v) into the drone together with the necessary camera rotations and photography.

5. Results

The model and procedure described above leads to the following results.

The first result is in an enclosed double-bay warehouse that does not have a satellite positioning system nor an IoT positioning system; the inventory can be checked automatically using a mid-range drone. It follows that the above model generally handles the movement of the drone in the aisles of a double-bay warehouse between the charging station (dock) and the corresponding starting compartment, and the last compartment and the dock. The outlined procedure provides a near-optimal route-planning method that can scan as many compartments as possible relative to the operating time of the drone and ensures a safe return to the starting point. Since the warehouse solution does not require a complete optimal solution, a well-tested OGA device can clearly be used.

The results can be divided into two parts: in the first part, a route-optimizing model was created. The characteristics of the warehouse were considered, such as the unified double-shelf system structure and the resolution of a compartment up to three parts. The drone was able to take photos from the middle of the aisle, so there was no need to move out of the bisecting plane towards the shelving systems. This was a simplifying condition discovered during the study in the logistics decenter. The drone was not responsible for finding the QR codes associated with the compartments, nor was it responsible for evaluating whether there were any goods placed in it. All these tasks can be easily performed by employing software to evaluate the photos, which could further accelerate the process. Another possible option is to take a picture of two opposite compartments in a row by rotating the camera, provided that it happens in a shorter time. This is influenced by specific values of the parameters. The first result obtained is the route optimization; this should be the procedure carried out prior to the launch of the drone.

The second result is on-the-fly control: after launching the drone from the dock, it receives instructions where to go, how to get there, and what activities to perform there (if camera rotation and photography are needed). The drone is instructed in real time for the next task for each position, while (of course) internal energy level monitoring is also performed.

6. Discussion

During our research of the literature, we came across several solutions related to optimal road planning. Most of them solve the problem with the help of neural networks. In their paper, Andrey V. Gavrilov and Artem Lenskiy propose a model for a new biologically inspired mobile robot navigation system. The novelty of their work is the combination of short-term memory and online neural-network learning using the event history stored in this memory. The neural network is trained with a modified error backpropagation algorithm that uses the principle of reward and punishment when interacting with the environment [18]. The robot navigation mechanism is one of the most challenging research topics in mobile robots, which requires the robot to find the right path and travel from its current position to the target position without encountering obstacles. In their paper,

Cheng and Cheng use the intelligent method of reinforcement learning to find a solution to the abovementioned problem. It considers the distances detected by a laser beam and the relative movement angle as the input of the neural network model, and the action posture of the robot as the output. This neural network model is trained by a deep Q-learning network (DQN) algorithm through positive and negative feedback rewards defined by task-specific learning goals. In this sense, the trained model helps the robot determine the appropriate steps to take in each state to safely reach the target without manual intervention. According to the results achieved on the simulation platform, the trained neural network model successfully moves the robot from a random starting point to a random destination, which proves the effectiveness of the DQN algorithm in the field of robotic navigation. [19] These models could also be applied to our problem, but it should be considered that the drone must not move in a plane, but in space. This would greatly complicate the solution. In fact, the solution we provide provides a simpler, more efficient solution. In particular, the application of neural networks would have required many examples, which we did not have at our disposal.

Another task from the research was to create an image processing and inventory analysis and a recording software that can be used for the task. The first task in Image Processing is to recognize the QR code itself and to identify the specific compartment. The identification of the compartment is provided by the routing model, and the route itself must be known to the processing application. Hence, automatically received photos are taken according to the route activities. The shelf system and the compartment where the photo was taken can be assigned to each photo by the application. This is a simple synchronization task.

The processing of the images is not very complicated, as most QR code readers are able to validate, and of course, interpret the data in the code. However, the fact that a compartment may be divided into several sub-compartments in a manner not previously known, so that several QR codes may have to be recognized, complicates the situation. It is also important that the QR code of another compartment should not be included in the photo and should not be handled by the recognizing application either. This can be easily prevented, as the shelf system is uniformly colored in most logistics centers—there were blue columns and red shelves in the warehouse we examined. With the help of these colors, the cells can be delimited, and the division can be determined based on the expected size by linear image processing. In other words, based on the image, one can see how many QR codes to search for, which can be easily carried out with the QR code search procedure. (Focusing was included into the shooting time of the drone).

The final processing application checks whether the actual unit load in the given compartment meets the requirements based on the position and QR code and indicates the compliance or deviations to the warehouse operator accordingly.

7. Conclusions

In this paper, we presented a model of a drone-driven inventory control solution. As a result of the method and with the help of the implemented application, the time of inventory testing can be greatly reduced, and the task can be performed with the help of an average human resource. A task that had once taken several days to complete can now be completed in about two days with this solution. In addition, the platform solution currently used, which endangers personal safety, can be avoided. It is obvious that in the case of the entire warehouse, the use of several drones can further reduce this time, and erroneous picking due to human errors can be eliminated by frequent inspections, so that the use of drones can also reduce the resulting losses.

Determining location based on QR code recognition is not a viable option, and one of the best known and most widely used methods is the Viola–Jones object detection framework. The mentioned framework provides an efficient way to focus the detection process on specific parts of the image. Based on our solution design, the initial state already shows a possible solution, and the intermediate states will always be possible solutions and

a good GA procedure is sufficient to apply a parameter reduction solution. In reality, there are many “good” paths to choose from. However, our tests show that the path depends significantly on the relationship between the underlying data.

At the beginning of our article, a full parameter exploration was performed. The range of data was defined with the help of practitioners, and then the basic collected data were grouped according to their role. Once the required parameters had been selected, it was important to record their dimensions. Next, we created a parametric model of the warehouse and the factory parameters of the drone to be used, improved with the data we had found. We created a manageable, simplified motion and speed. The next step was to develop a practical positioning strategy. Then, we created a mathematical model of the operation and assigned the theorems and determined environment needed for optimization.

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

Funding: This research project was funded by Budapest Business School Research Fund.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

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. Bányai, T.; Illés, B.; Gubán, M.; Gubán, A.; Schenk, F.; Bányai, A. Optimization of Just-In-Sequence Supply: A Flower Pollination Algorithm-Based Approach. *Sustainability* **2019**, *11*, 3850. [CrossRef]
2. An, R.; Guo, S.; Zheng, L.; Hirata, H.; Gu, S. Uncertain moving obstacles avoiding method in 3D arbitrary path planning for a spherical underwater robot. *Robot. Auton. Syst.* **2022**, *151*, 104011. [CrossRef]
3. Fülöp, M.T.; Gubán, M.; Gubán, A.; Avornicului, M. Application Research of Soft Computing Based on Machine Learning Production Scheduling. *Processes* **2022**, *10*, 520. [CrossRef]
4. Pehlivanoglu, Y.V.; Pehlivanoglu, P. An enhanced genetic algorithm for path planning of autonomous UAV in target coverage problems. *Appl. Soft Comput.* **2021**, *112*, 107796. [CrossRef]
5. Benes, F.; Stasa, P.; Svub, J.; Alfian, G.; Kang, Y.-S.; Rhee, J.-T. Investigation of UHF Signal Strength Propagation at Warehouse Management Applications Based on Drones and RFID Technology Utilization. *Appl. Sci.* **2022**, *12*, 1277. [CrossRef]
6. Gerwen, J.V.-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.* **2021**, *71*, 961–974. [CrossRef]
7. Rhiat, A.; Chalal, L.; Saadane, A. A Smart Warehouse Using Robots and Drone to Optimize Inventory Management. In *Proceedings of the Future Technologies Conference (FTC) 2021*; Springer: Berlin/Heidelberg, Germany, 2021; Volume 1, pp. 475–483.
8. 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.* **2021**, *18*, 582–591. [CrossRef]
9. Szegedi, Z. *Raktározáslogisztika*; AMEROPA Kiadó: Budapest, Hungary, 2010.
10. Rjeb, A.; Gayon, J.-P.; Norre, S. Sizing of a homogeneous fleet of robots in a logistics warehouse. *IFAC-PapersOnLine* **2021**, *54*, 552–557. [CrossRef]
11. eyesee-drone.com. Available online: <https://eyesee-drone.com/eyesee-the-inventory-drone-solution/> (accessed on 10 March 2022).
12. Jácome, R.N.; Huertas, H.L.; Procel, P.C.; Garcés, A.G. Fuzzy Logic for Speed Control in Object Tracking Inside a Restricted Area Using a Drone. In *Developments and Advances in Defense and Security*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 135–1456.
13. Paredes, J.A.; Álvarez, F.J.; Aguilera, T.; Aranda, F.J. Precise drone location and tracking by adaptive matched filtering from a top-view ToF camera. *Expert Syst. Appl.* **2019**, *141*, 112989. [CrossRef]
14. Li, F.; Zlatanova, S.; Koopman, M.; Bai, X.; Diakité, A. Universal path planning for an indoor drone. *Autom. Constr.* **2018**, *95*, 275–283. [CrossRef]
15. Sacramento, D.; Pisinger, D.; Ropke, S. An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones. *Transp. Res. Part C: Emerg. Technol.* **2019**, *102*, 289–315. [CrossRef]
16. Dhein, G.; Zanetti, M.S.; de Araújo, O.C.B.; Cardoso, G., Jr. Minimizing dispersion in multiple drone routing. *Comput. Oper. Res.* **2019**, *109*, 28–42. [CrossRef]

17. Morocho-Cayamcela, M.E.; Lim, W.; Maier, M. An optimal location strategy for multiple drone base stations in massive MIMO. *ICT Express* **2021**, *8*, 230–234. [\[CrossRef\]](#)
18. Gavrilov, A.V.; Lenskiy, A. Mobile Robot Navigation Using Reinforcement Learning Based on Neural Network with Short Term Memory. In *ICIC 2011: Advanced Intelligent Computing*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 210–217.
19. Cheng, C.; Chen, Y. A Neural network based mobile robot navigation approach using reinforcement learning parameter tuning mechanism. In *Proceedings of the 2021 China Automation Congress (CAC)*, Beijing, China, 22–24 October 2021.