Article

Non-Traditional Layout Design for Robotic Mobile Fulfillment System with Multiple Workstations

Xiuqing Yang [1], Xinglu Liu [2,*] , Lijuan Feng [3] , Jianquan Zhang [1] and Mingyao Qi [3]

[1] The Civil Aviation Logistics Technology Co., Ltd., The Second Research Institute of Civil Aviation Administration of China, Chengdu 610041, China; yangxiuqing@caacetc.com (X.Y.); zhangjianquan@caacetc.com (J.Z.)

[2] Intelligent Transportation and Logistics Systems Laboratory, Tsinghua-Berkeley Shenzhen Institute, Shenzhen 518055, China

[3] Research Center for Modern Logistics, Graduate School at Shenzhen, Tsinghua University, Shenzhen 518055, China; fenglj16@mails.tsinghua.edu.cn (L.F.); qimy@sz.tsinghua.edu.cn (M.Q.)

* Correspondence: liuxl18@mails.tsinghua.edu.cn

Abstract: This paper studies the layout design of a robotic mobile fulfillment system with multiple workstations. This is a parts-to-picker storage system where robots hoist pods and bring them directly to the workstations for stationary pickers to retrieve required items. As few research efforts have focused on determining the optimal locations of workstations in such systems, we develop an integer programming model to determine the location of workstations to minimize the total traveling distance of robots. In addition, we investigate the near-optimal workstation location patterns (i.e., some general workstation configuration rules) in the context of both traditional and flying-V layouts. A series of experiments led to the following findings: (1) the flying-V layout can save 8∼26% of travel distance compared with the traditional layout, and the sacrifice of space use is only 2∼3% for medium or large warehouses; (2) instead of solving the optimization model, the proposed $2n$ rule and $n + 1$ rule are simple and easily implemented ways to locate workstations, with travel distance gaps of less than 1.5% and 5% for traditional and flying-V layouts, respectively; and (3) the "optimal" cross-aisle angle (i.e., θ) in flying-V layout can be set as large as possible as long as the cross-aisle intersects the left or right edge of the warehouse.

Keywords: flying-V layout; workstation location; integer programming; parts-to-picker; robotic mobile fulfillment system; traditional layout; warehouse design

1. Introduction

Driven by the global trade expansion and the world economy reformation, the logistics sector has grown rapidly over the past few decades [1], triggering a tremendous increase in freight transportation (e.g., road transportation, [2]; maritime transportation, [3]; air freight transportation, [4]), and enabling growth in cargo volumes handled at different types of terminals (i.e., warehouses, [5]; rail terminals, [2]; cross-docking facility, [6]; marine terminals, [4], etc.). Unfortunately, this also causes negative impacts on the environment to some extent (e.g., carbon emission [7]), and thus, enhancing the efficiency of freight transportation and creating a more sustainable logistics system is of great importance for both enterprises and the society. Among the aforementioned terminals, warehouses are perhaps the ones most relevant to the e-commerce community. Typically, warehouses are labor-intensive that include receiving, storage, order-picking and shipping processes. Among these, the order-picking process has been identified as the most labor-intensive and costly activity for the operation of warehouses (De Koster et al. 2007, [8]). To curb operational costs and increase throughput capacity of warehouses, more and more companies adopt new storage and order-picking technologies, especially e-commerce businesses with fast turnover of goods and strong demand fluctuations. For instance, Kiva System,

also called Amazon Robotics, saves pickers a lot of running back and forth fetching items from the shelves (Mountz, 2012, [9]). The implementation of this kind of system can be seen in Amazon.COM, Alibaba.COM and JD.COM China. This type of warehouse system is referred to as a robotic mobile fulfillment system (RMFS) (Lamballais et al., 2017, [10]), where robots hoist movable storage pods (i.e., inventory pods) and deliver them directly to the workstations for stationary pickers to retrieve required items. Unlike the traditional order-picking system, which requires pickers travel to each storage rack, this emerging parts-to-picker system allows the picker to concentrate on picking and packing orders (Boysen et al. 2017, [11]).

There remain numerous factors impacting the order-picking process in a warehouse, including warehouse layout, storage strategy, zoning, batching and routing method (see De Koster et al. 2007 [8] and Yu 2009 [12]). The warehouse layout is one of the most important components [13], which will affect the daily operations (Dukic, G and Opetuk, T, 2012 [14]; Roodbergen et al. 2015 [15]). Typically, in a traditional warehouse, picking aisles must be straight and parallel to each other; cross aisles should meet picking aisles at right angles, which forces pickers to travel rectilinear distances to picking locations. However, Gue and Meller (2009) [16] break these rules and propose two novel aisle designs for the unit-load warehouse with a single pickup and deposit point (here referred as a workstation), i.e., so-called flying-V and fishbone, which can reduce expected travel distance by about 8~12% and more than 20%, respectively. Gue and Meller may be the first to study non-traditional warehouse design after White (1972) [17] put forward the optimum warehouse design with radial aisles. They extended their study to facilitate multiple pickup and deposit points using a flying-V layout (Gue et al. 2012 [18]), which can save 3~6% of the travel distance compared with a traditional layout. Inspired by these research efforts, we think it would be very interesting to investigate the performance of new layout designs under the context of a robotic mobile fulfillment system. It is worth mentioning that another similar scenario to the aforementioned issue is container picking in port yard systems and their layout design (see [19,20] for more details). Typically, in such systems, containers are carried by ground trolleys, shipped to quay cranes, and finally loaded to vessels. These objects mentioned above (i.e., containers, ground trolleys, and quay cranes) can be viewed as inventory pods, robots, and workstations in RMFS, respectively. There remain several essential differences between the two, but the one of most relevance to this work is the workstation location restrictions. Specifically, the quay cranes (i.e., workstations in this work) are required to be deployed on the same side of the yard (i.e., the side close to the vessels) in the yard system due to its structural limitations, whereas such restrictions do not exist in RMFS.

As multiple workstations could be employed in medium or large warehouses, whereas according to recent survey papers ([5,21]) and our knowledge, the only two published literature in which the layout aspects of RMFS were mentioned is Lamballais, Roy, and De Koster, (2017) [10] and Wu et al., (2020) [22]. Lamballais et al. maintain that the locations of the workstations in warehouses directly affect the system throughput. Wu et al. estimate the warehouse performance under various layout configurations. Unfortunately, these analyses are based on several given workstation layouts (multiple workstations), more specifically, four fundamental layout setups in reference [10] and seven layout scenes in reference [22] are pre-defined respectively, and then served as input settings for their proposed model, and afterwards, the performance of these layouts are further evaluated in a simulation manner. This seems exactly a main drawback of the above research that related to our work, i.e., they do not formulate workstation layout as decision variables explicitly and further improve the performance of RMFS from a warehouse design perspective. In summary, most relevant published literature performs single workstation assumption ([16,23–26]), or consider multiple ones but tend to treat the positions of the workstations as pre-defined information ([10,18,27–31]). As we will verify in this paper, optimal workstations are not located symmetrically on the top and bottom edges of the warehouse, which is counter-intuitive to our experiences.

In this paper, we apply non-traditional warehouse layout design to optimize the location of multiple workstations. Specifically, the flying-V layout is adopted for a RMFS to minimize the total travel distance of all robots in the system. We are going to address the following research questions:

1. How to appropriately locate the workstations in traditional and flying-V layouts?
2. Whether there exist any patterns under the optimal layout of workstations?
3. Is flying-V a good substitute of traditional layout?

To answer these questions, we first develop the expression of travel distance from any pod to any single workstation, for both traditional and non-traditional layouts; then, we formulate an integer programming model to optimize the location of multiple workstations. We demonstrate the proposed model can be solved by a commercial solver (for example, CPLEX) in a very short time even for very large warehouses in the e-commerce industry. We also carry out comprehensive numerical experiments to compare the performance of different warehouse layout types (traditional and flying-V) and different warehouse configurations. Our study makes the following contributions:

1. To the best of our knowledge, we might be the first to investigate a non-traditional layout for a robotic mobile fulfillment system.
2. We propose an integer programming model to determine the optimal location of workstations.
3. We propose two general rules that can provide an approximately optimal workstation location solution with a less than 5% gap.

The remainder of this paper is organized as follows: Section 2 reviews relevant literature; Section 3 presents the expression of the travel distance from any pod to any workstation; and Section 4 presents the workstation location model. Section 5 reports the numerical experiments and illustrates the findings, and finally, in Section 6, conclusions are summarized, and future research directions are identified.

2. Literature Review

This section reviews three aspects of the related research work. First, we will start from the research on traditional layout designs, followed by research on non-traditional aisle designs. We shall also review the design and operational studies in the context of RMFS. Table 1 summarizes the related literature in terms of factors including the topic, aisle type, warehouse type, storage policy, objective, methodology and decisions.

Table 1. Literature overview of warehouse design problem.

Reference	Topic	Aisle Type	Warehouse Type	Storage Policy	Objective	Method	Decision/Output	MW	DW
Caron et al. (2000) [32]	layout design	traditional	MOP	COIR	min ETD	analytics	Approach to access different layouts	×	×
Pohl et al. (2009) [33]	layout design	traditional	UL	R	min ETD	analytics	Comparison of three common warehouse designs	×	×
Roodbergen and Vis (2006) [34]	layout design	traditional	MOP	R	min ETD	analytics	Optimal layout and estimation of TD	×	×
Roodbergen et al.(2008) [35]	layout design	traditional	MOP	R	min ETD	analytics	Optimal layouts with any number of blocks and aisles	×	×
Gue and Meller (2009) [16]	layout design	flying-V, Fishbone	UL	R	min ETD	analytics	cross-aisle intersection points	×	×
Gue et al. (2012) [18]	layout design	flying-V	UL	R	min ETD	analytics	cross-aisle intersection points	√	×
Cardona et al. (2012) [24]	layout design	fishbone	UL	R	min ETD	analytics	slope of the diagonal cross-aisle	×	×
Pohl et al.(2011) [36]	layout design and storage policy	flying-V, fishbone	UL	R, TB	min ETD	analytics	Optimal layout in TB storage policy	×	×
Öztürkoğlu et al.(2012) [25]	layout design	chevron, leaf, butterfly	UL	R	min ETD	analytics	cross-aisle angle	×	×
Öztürkoğlu et al.(2014) [29]	layout design	irregular type with at most two cross aisles	UL	R	min ETD	heuristics	cross-aisle angle	√	×
Çelk et al. (2014) [37]	layout design and storage policy	Flying-V, fishbone	UL	R, TB	min ETD	heuristics	performance evaluation	×	×
Bortolini et al. (2015) [23]	layout design	diagonal	UL	R	min ETD	analytics	performance evaluation	×	×

Table 1. *Cont.*

Reference	Topic	Aisle Type	Warehouse Type	Storage Policy	Objective	Method	Decision/Output	MW	DW
Venkitasubramony (2016) [26]	layout design	fishbone	UL	R, TB, CB	min ETD	analytics	performance evaluation	×	×
Boysen et al.(2017) [11]	order-picking sequence	traditional	RMFS	TB	min NRC	MIP	order-picking sequence	×	×
Li et al. (2017) [38]	order-picking sequence	traditional	RMFS	D	min RTT	IP	order-picking sequence	×	×
Yuan et al.(2019) [30]	storage policy	traditional	RMFS	R, TB, CB	min ETD	analytics	Fluid model to analyze storage policy performance	√	×
Merschformann et al. (2018) [27]	operational rules	traditional	RMFS	R, D, CB, etc.	-	simulation	order assignment, pod selection and storage policies	√	×
Zou et al. (2017) [31]	operational rules	traditional	RMFS	R	min RTT	analytics	assignment rules evaluation	√	×
Lamballais et al.(2017) [10]	performance evaluation	traditional	RMFS	TB	max TP	analytics	warehouse layout and robot zoning evaluation	√	×
Wu et al.(2020) [22]	layout design	traditional	RMFS	R, ZB	max TP	SOQN-based simulation	warehouse layout and parameter configuration	√	×
Jin et al.(2020) [39]	layout performance evaluation	traditional (compact storage)	RMFS	R	-	SOQN-based simulation	warehouse layout and parameter configuration	√	×
this work	layout design	traditional, Flying-V	RMFS	R	min ETD	IP	optimal location of workstations	√	√

MOP, manual order-picking; UL, unit-load warehouse; R, random storage; D, dedicated storage; TB, turnover-based storage policy; CB, class-based storage policy; ZB, zone-based storage policy; TP, throughput; NRC, number of rack changes; ETD, expected travel distance; RTT, retrieval throughput time; SOQN, semi-open queueing network model; IP(MIP), (mixed) integer programming; MW, multiple workstations (simulation or compare different pre-defined layouts); DW, decision of workstations.

2.1. Traditional Layout Designs

As Gu et al. (2007, 2010) [40,41] point out, warehouse design involves five major decisions, including overall warehouse structure, sizing and dimensioning, department layout, equipment selection, and operational strategies, among which layout and operational strategies are mostly addressed owing to the convenience of quantitative analysis. Warehouse layout problems usually focus on exploring the system parameters affecting optimal design, including the total length of picking aisles and the number of picking aisles and blocks (Caron, 2000 [32]; Roodbergen and Vis, 2006 [34]; Roodbergen et al. 2008 [35]). Please note that most of the research assumes only one workstation.

The three most common layouts in traditional design problems are shown in Figure 1 (similar description see Pohl et al. 2009 [33]). Layout *A* has picking aisles perpendicular to the front wall but without a cross-aisle inserted into the picking space. In layout *B*, picking aisles are perpendicular to the front wall with a middle cross-aisle, while in layout *C*, picking aisles are parallel to front wall with a central cross-aisle. Obviously, the layout optimization problem is about the arrangement of storage locations and aisles, and is often evaluated by expected travel distance for pickers. Pohl et al. (2009) [33] developed expected travel distance expressions based on any two locations in the storage area for dual-command operations and used them to analyze the three common warehouse designs. Their results indicated that layout *C* is the best design, which is least commonly found in practice. Caron et al. (2000) [32] present an analytical approach for layouts *A* and *C* (with workstations at central and corner points) to calculate the expected travel distance with *N* picking locations using a traversal routing strategy. Both COI-based and random storage policies are considered. The difference in travel distance between layout *A* and *C* may be greater than 60%. Moreover, the results show that layout decisions seem to be strongly affected by decisions concerning batching and storage policies. For layout *B* and its variants (with more than one block), Roodbergen and Vis (2006) [34] developed a model capable of finding the best layout for one block order-picking area, with S-shaped and largest-gap routing policies. Based on this work, Roodbergen et al. (2008) [35] extended the research and formed a layout optimization model that can consider layouts with any number of blocks and aisles using a statistical estimate for average travel distances with random storage and S-shape routing. All these layouts consider only one workstation (which is also referred to as a pickup and deposit point, P&D), and Roodbergen and Vis (2006) [34] have proven that the optimal location of the P&D is in the middle of the front cross-aisle. Owing to the characteristics of the manual order-picking system, such as pickers having to visit more than one storage location during a tour, the S-shaped or traversal routing policy is commonly used when calculating travel distance.
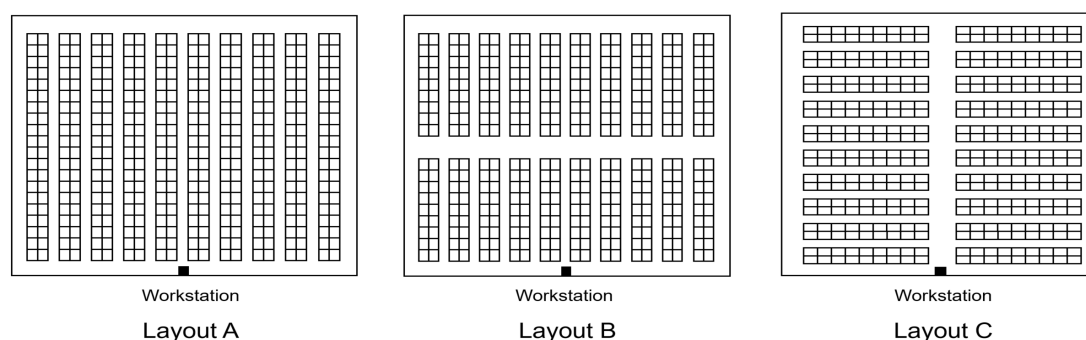

Figure 1. Three most common warehouse aisle layouts.

2.2. Non-Traditional Layout Designs

Among existing research on non-traditional layout design, flying-V and fishbone are the most often addressed (Gue and Meller, 2009 [16]; Gue et al., 2012 [18]; Cardona et al., 2012 [24]; Pohl et al., 2011 [36]). The flying-V design has a V-shaped cross-aisle with vertical

picking aisles (Figure 2, left). The fishbone design features picking aisles with different angles (Figure 2, right). These various types of layouts aim to reduce travel distance while slightly sacrificing space use. Since the aisle configurations under fishbone layout (multiple horizontal aisles and multiple vertical aisles) differ significantly from the traditional one (all aisles are deployed vertically), it seems less practical to investigate the differences of optimal workstations layout between fishbone layout and traditional layout. In this work, we only consider flying-V layout (only two more angled cross aisles than traditional layout) due to its higher aisle configuration similarity to the traditional layout, focusing on determining the optimal deployment of multiple workstations and comparing the results with those of the traditional cases. Moreover, to simplify the travel distance calculation, we assume that the angled cross aisles in flying-V warehouses are straight rather than curved.
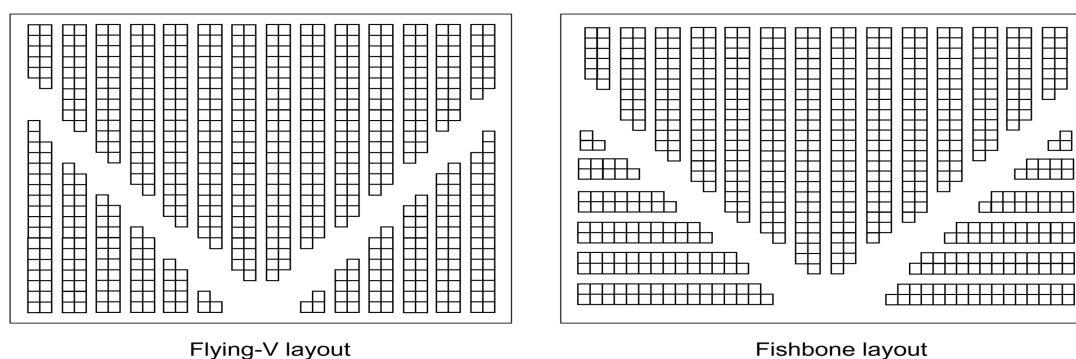

Flying-V layout   Fishbone layout

Figure 2. The flying-V (left) and fishbone layout (right).

Gue and Meller (2009) [16] first proposed flying-V and fishbone layouts and estimated the expected travel distance under single and multiple workstations. Results show that flying-V layout promises to reduce expected travel distance by about 8–12%, and fishbone can save more than 20%. In both single and multiple workstations cases, the decision variables are intersection points of the cross-aisle, while workstations are fixed and are not treated as decision variables. A continuous integral approach is applied for distance calculation with the assumption of uniform pick density (random storage). Similar research can refer to Cardona et al. (2012) [24] and Mesa et al. (2017) [28]. Öztürkoğlu et al. (2012) [25] put forward the chevron, leaf, butterfly layout and determined the corresponding optimal designs. Then, Öztürkoğlu et al. (2014) [29] adopted a discrete graph-based approach to compute the expected distance, which is different from former continuous distance models, and they applied particle swarm optimization (PSO) methods to generate some irregular aisle types. The results also suggested that the potential benefit of alternative aisle designs depends highly on the number and locations of workstations. All the above-mentioned literature focus on determining optimal cross-aisle parameters (i.e., angle of cross aisles, and intersection points positions of angled cross-aisle and vertical aisle) to minimize total expected travel distance.

2.3. RFMS Design and Operation

Following the earlier work about Kiva Systems (Enright and Wurman, 2011 [42]), most of the recent research about RMFS focuses on operational issues including: order and rack sequencing (Boysen et al., 2017 [11]; Li et al., 2017 [38]; Merschformann et al., 2019 [27]; Kim et al., 2020, [43]; Xie et al., 2021, [44]), workstation assignment (Zou et al., 2017 [31]; Xie et al., 2021, [44]), storage policy (Yuan et al., 2019 [30]; Merschformann et al., 2019 [27]; Kim et al., 2020, [43]; Lamballais et al., 2020, [45]), and traffic planning (Wurman et al., 2007 [46]; Herrero-Pérez, D and Martínez-Barberá, H, 2011 [47]; Qi et al., 2018 [48]). Boysen et al. (2017) [11] propose a mobile robot-based order-picking problem (MROP) to determine the order sequence that was handled in a workstation as well as the sequence of racks that feed the workstation, whereas no discussion on layout was conducted. Li et al.

(2017) [38] studied the similar problem using a single workstation by formulating it as an integer programming model. Yuan et al. (2019) [30] developed a linear travel distance function to discuss several typical storage policies. Their study considers multiple workstations that are located around the storage area, but the positions of workstations are pre-defined. Merschformann et al. (2019) [27] used discrete event simulations to evaluate the RMFS performance in both the pick and replenishment process by evaluating multiple decision rules related to order assignment, pod selection and storage assignment problems. The workstation layout problem is pointed out but not addressed. Xie et al. (2021) [44] investigate the optimal operational policies with splitable orders in RMFS, in which the decisions that assignment of pods to workstations and orders to workstations are involved. They claim that the system performances are significantly improved (e.g., increasing throughput by 46%) if parts of an order are allowed to be picked at different stations. Although some decisions that are related to this work are considered (i.e., pod-to-workstation assignment), the workstation layout decisions are still not mentioned.

There is also some research addressing design problems on a strategic level, typically by analytic methods. Zou et al. (2017) [31] used queue theory to evaluate the retrieval throughput time under different policies by which to assign pods to workstations. The paper analyzed the optimal shelf block size combined with an optimal length-to-width ratio. Lamballais et al. (2017) [10] applied queue theory again to estimate maximum order throughput, average order cycle time, and robot use. Through a set of numerical experiments, they found that the warehouse performance is affected by the location of the workstations around the storage area, as well as the length-to-width ratio. However, both papers treat the locations of workstations as a fixed position instead of considering them as decision variables.

Storage policy is obviously important to the system performance, which determines which location is assigned for a returning pod and thus influences the total travel distance of robots. According to Yuan et al. (2019) [30], there are generally four storage policies, namely full-turnover-based storage, class-based storage, closest-open-location storage, and random storage. These policies have advantages and disadvantages in terms of travel distance and space use, and depending on application context. This paper adopts the random storage policy, in which items are stored randomly on pods, and pods are returned to randomly assigned storage locations. It can lead to a high space use, since all storage locations are available to all storage units. As we can see from Table 1, Zou et al. (2017) [31] also used this policy, and Merschformann et al. (2018) [27] and Yuan et al. (2019) [30] compare the performance under random storage with other storage policies. Kim et al. (2020) [43] study the item assignment problem in the RMFS, aiming to maximize the total similarity scores of items in each inventory pod and thus improving the picking efficiency. Lamballais et al. (2020) [45] discuss the pod's inventory allocation problem from a more macro perspective, i.e., the replenishment level per pod, etc.

We notice that most recent research focus on warehouse performance in a local perspective. To the best of the authors' knowledge, few published studies focus on the workstation location, which is a fundamental and strategic problem for a RMFS. The only two highly relevant papers from the literature are Lamballais, Roy, and De Koster, 2017 ([10]) and Wu et al. 2020 ([22]). Nevertheless, they analyze the impacts of workstation locations by comparing the performance of several pre-defined layouts in a simulation-based manner, and fail to formulate workstation locations as decision variables to avoid local optimal. To fill this research gap, therefore, we explicitly consider the locations of multiple workstations as decision variables under traditional layout and flying-V layout in RMFS, formulate workstation optimization problem as an integer programming, and explore related managerial insights in this work.

3. Problem Description and Distance Expression

The RMFS addressed in this paper is a parts-to-picker semi-automated storage system that is emerging in the e-commerce industry. The goods are stored on inventory pods,

where robots are used for transport and pickers concentrate on picking and packing orders at workstations. Typically, a shared storage policy is implemented in that each pod consists of shelves with different stock keeping units (SKUs), while each item may exist in more than one pod. An order may have single or multiple lines, each identifying a SKU and its quantity. We assume the whole warehouse is a rectangle zone and the workstations are placed on the top and bottom boundaries of this zone. This paper aims to optimize location of the workstation to minimize the total robot travel distance. As this is a strategic decision rather than an operational one, we chose to ignore some details and conduct our research with the following assumptions:

1. A pod is always assigned to the nearest workstation from itself among all workstations, thus the work balance among different workstations is not considered, and situations in which more workstations need the same pod are neglected;
2. A random storage assignment policy is adopted according to which pod will be returned to a randomly selected location after picking in the workstation, and hence we treat the possibility of transporting a pod from any location to any workstation as roughly the same;
3. The traffic congestion, conflict and deadlock due to multiple robots in a limited space is ignored, enabling a shortest pathway for the metric of travel distance.
4. The warehouse is modeled as a discrete graph space that consists of individual pod storage units (we refer as pods for conciseness), workstations, as well as the paths that connect them. Please note that the insertion points of cross aisles (i.e., the intersection of cross aisles and warehouse boundaries) in flying-V layout are also included. Additionally, we assume the robots travel in the centerline of aisles. In the following two sections, we will deduce the distance expression for both traditional and flying-V layout based on a graph-based network.

For assumption 1, suppose the order consists of multiple lines items, each pod is assigned to the nearest workstation, which implies that different pods for the same order may end up at different workstation and therefore further processing is required. This proves reasonable according to Xie et al. (2021) [44], in which the results reveal that considerable improvements of several performance metrics are obtained in RMFS if parts of an order are allowed to be picked at different stations. Assumption 2 also seems rational, because Boysen et al. (2018) [49] and Lamballais, Roy and de Koster (2017) [45] have demonstrated that same SKU are spread over multiple inventory pods, i.e., so-called mixed-shelves storage (can be viewed as random storage), dramatically boosting the picking process.

3.1. Travel Distance Expression in a Traditional Layout

Following the same assumption as [10], we specify that each pod must take the shortest path if it will be delivered to a candidate workstation. In addition, the distance equations under traditional layout are similar to those in [10]. Figure 3 illustrates a sketch map of a traditional warehouse layout. The number of rows and columns are L and N respectively. Without loss of generality, N is supposed to be an even number and $N \geqslant 4$. We define a coordinate system where the origin locates in the center of the rectangle bottom edge. As shown in Figure 3, a pod is denoted by (i, j), which represents that the pod is in row $j, j \in [1, L]$, column $i, i \in \left[-\frac{N}{2}, \frac{N}{2}\right]$. The width of each picking aisle is w_a, and the width of each pod is w_p. We only consider two cross aisles at the top and bottom edges with width w_e. We assume that there are $|\mathcal{R}|$ candidate workstations (stars in Figure 3), and each is denoted by (x_r, y_r), $r \in \mathcal{R}$ (where $\mathcal{R}$ is the set of candidate workstations). The center

of a workstation is $\frac{1}{2}w_p$ away from the edge. Each pod is associated with a point by its geometric center, denoted by (x_{ij}, y_{ij}) in the coordinate system that:

$$x_{ij} = \left(\frac{i}{|i|}\right) \cdot \left[\frac{1}{2}w_a + \left(\lceil\frac{|i|}{2}\rceil - 1\right)w_a + \left(|i| - \frac{1}{2}\right)w_p\right]$$
$$y_{ij} = \left(j - \frac{1}{2}\right)w_p + w_e \tag{1}$$

where $\lceil \rceil$ refers to the ceiling function. We define the travel distance error caused by whether the column index is an odd number or an even number as

$$e_i = [(|i| + 1) \bmod 2](w_p + w_a) \tag{2}$$

If pod (i, j), located at (x_{ij}, y_{ij}), is retrieved and delivered to the workstation r, then the distance can be measured by the following expression:

$$d_{ijr} = |x_{ij} - x_r| + e_i + |y_{ij} - y_r| \tag{3}$$

We notice that the error term, e_i, exists in the travel distance estimation for 50% of the pods in the warehouse, but its impact on the calculation results is marginal and hence can be neglected. Therefore, we rewrite Equation (3) by

$$d_{ijr} = |x_{ij} - x_r| + |y_{ij} - y_r| \tag{4}$$
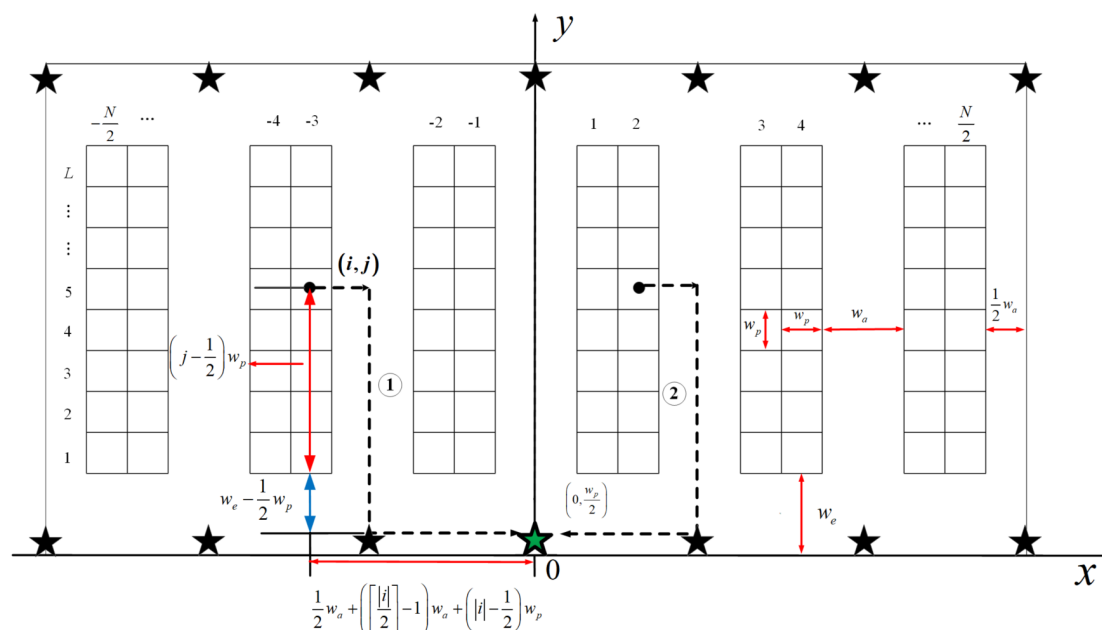
Figure 3. Sketch of a traditional warehouse layout.

3.2. Travel Distance Expression in Flying-V Layout

Distance equations seem exceedingly complicated under flying-V settings due to the introduction of angled cross-aisle. Although the distance expressions in flying-V layout warehouses with single(double) workstation(s) are provided in several existing literature, e.g., [16,18,23–26,28], they seem not applicable under multiple workstation cases. As we will elaborate below, the distance equations vary with the area where the pod and targeted workstation are located. To the best of our knowledge, it is the first attempt that the shortest distance equations under flying-V layout with multiple workstations are provided (i.e., Equations (11)–(16) in this section).

In the flying-V layout, two cross aisles are inserted into the warehouse, as shown in Figure 4. The inserted cross aisles begin at the bottom center, extending to the right-most and left-most edges with angle θ and $\pi - \theta$, respectively. Owing to its complexity, we will first demonstrate the expression of robot travel distance with a single workstation located at the center of the bottom edge. There exist two possible candidate paths if we try to deliver pod (i, j) to the workstation (x_0, y_0) under a flying-V layout, as shown in Figure 4 (herein, we have $x_0 = 0, y_0 = \frac{1}{2}w_p$ according to the description in Section 3.1) (Similar descriptions of possible paths in flying-V layout see [50]). Path one (i.e., the solid red line) heads down (or up) from the current position along the corresponding vertical aisle, reaching the angled cross-aisle, and then travels along the angled cross-aisle, and finally arrives at the workstation. Path two, which is identical to the delivery path in a traditional warehouse, moves down along the corresponding vertical aisle, travels horizontally along the bottom aisle, and finally reaches the workstation. We assume that each pod is delivered from its current position to the workstation along the shortest path. Again, we ignore the error caused by the even column index. Thus, the distance of path one can be given by (The exact distance formula of path one is $\tau_0 = \frac{1}{2}\left(w_p + w_a\right) + \frac{|x_{ij}| - \frac{1}{2}\left(w_p + w_a\right)}{\cos\theta} + \left|\left(|x_{ij}| - \frac{1}{2}\left(w_p + w_a\right)\right)\tan\theta - y_{ij}\right|$.

To simplify the calculation, we approximate τ_0 by $d_{ij}^{p_1} = \frac{|x_{ij}|}{\cos\theta} + ||x_{ij}\tan\theta| - y_{ij}|$. Such approximation is reasonable, because the difference between two formulas can be given by $\epsilon = \tau_0 - d_{ij}^{p_1} = \frac{1}{2}\left(w_p + w_a\right)\left(\tan\theta + 1 - \frac{1}{\cos\theta}\right)$, where $\theta \in \left[\frac{5}{36}\pi, \frac{13}{36}\pi\right]$, and $w_p = 1, w_a = 2$ in this work. Please note that $\epsilon \in [0.54, 1.17]$ under the above conditions, which is negligible compared to τ_0, and thus, we replace τ_0 with $d_{ij}^{p_1}$. As for multiple workstation cases, the above conclusion still holds after similar analysis. For other distance calculations, we also adopt similar approximation method, see Table 2):

$$d_{ij}^{p_1} = \frac{|x_{ij}|}{\cos\theta} + ||x_{ij}\tan\theta| - y_{ij}|. \tag{5}$$

Since the trajectory of path two in flying-V warehouses is exactly the same as the delivery path in traditional warehouses, we can derive its distance expression by Equation (4), i.e.,

$$\begin{aligned} d_{ij}^{p_2} &= |x_{ij} - x_0| + |y_{ij} - y_0| \\ &= |x_{ij}| + y_{ij} - \frac{1}{2}w_p \end{aligned} \tag{6}$$

Then the shortest distance from pod (i, j) to workstation (x_0, y_0) is

$$d_{ij} = \min\{d_{ij}^{p_1}, d_{ij}^{p_2}\} \tag{7}$$

Suppose we need to deploy $K(1 \leqslant K \leqslant |\mathcal{R}|)$ workstations in the warehouse. To facilitate the calculation of distance between any pod and any candidate workstation, we divide the storage region into categories A, B, C and D (Figure 5). The candidate workstation set is given by

$$\begin{aligned} S_{\text{candidate}} = \{&S_1, S_2, S_3, S_4, A_1, \cdots, A_{\frac{N}{2}}, B_1, \cdots, B_{\frac{N}{2}}, \\ &C_1, \cdots, C_{\frac{N}{2}}, D_1, \cdots, D_{\frac{N}{2}}\} \end{aligned}$$

Table 2. Distance notations.

Notation	Description	Detailed Formula
l_a	length of the angled cross-aisle	$l_a = \frac{x_{\max}}{\cos\theta}$
$l_{p,a}$	required travel distance from pod (i,j) to the angled cross-aisle (i.e., the intersection of the angled cross-aisle and the corresponding vertical aisle) along the corresponding vertical aisle	$l_{p,a} = \left\lvert \left(y_{ij} - \lvert x_{ij} \rvert \tan\theta \right) \right\rvert$
l_{p,S_1}^a	required travel distance from pod (i,j) to candidate workstation S_1 along the angled cross-aisle	$l_{p,S_1}^a = \frac{\lvert x_{ij} \rvert}{\cos\theta}$
$l_{p,r}^a$	required travel distance from pod (i,j) to the column which candidate workstation r locates along with the angled cross-aisle	$l_{p,r}^a = \frac{\lvert x_{ij} - x_r \rvert}{\cos\theta}$
$l_{S_1,r}^a$	required travel distance from candidate workstation S_1 to the column which candidate workstation r locates along with the angled cross-aisle from candidate workstation	$l_{S_1,r}^a = \frac{\lvert x_r \rvert}{\cos\theta}$
$l_{p,r}^{bot}$	required travel distance for pod (i,j) travels from the angled cross-aisle (actually refers to the corresponding intersection) to candidate workstation r along with the vertical aisle which r locates (if the candidate workstation r locates at bottom side)	$l_{p,r}^{bot} = \lvert x_r \rvert \tan\theta$
$l_{p,r}^{a,top}$	required travel distance for pod (i,j) travels from the angled cross-aisle (actually refers to the corresponding intersection) to candidate workstation r along with the corresponding vertical aisle (if the candidate workstation r locates at the top side)	$l_{p,r}^{a,top} = y_{\max} - \lvert x_r \rvert \tan\theta$
l_{p,S_3}^a	required travel distance for pod (i,j) travels from its own position to candidate workstation S_3 along with the angled cross-aisle	$l_{p,S_3}^a = \frac{x_{ij} - x_{S_3}}{\cos\theta}, x_{ij} < 0$
l_{p,S_4}^a	required travel distance for pod (i,j) travels from its own position to candidate workstation S_4 along with the angled cross-aisle	$l_{p,S_4}^a = \frac{x_{S_4} - x_{ij}}{\cos\theta}, x_{ij} > 0$
l_p^{top}	required travel distance for pod (i,j) travels from its own position to top side along with the corresponding vertical aisle	$l_p^{top} = y_{\max} - y_{ij}$
l_p^{bot}	required travel distance for pod (i,j) travels from its own position to bottom side along with the corresponding vertical aisle	$l_p^{bot} = y_{ij} - y_{S_1}$
$l_{S_3}^{top}$	required travel distance from candidate workstation S_3 to top side along with the vertical aisle	$l_{S_3}^{top} = y_{\max} - y_{S_3}$
$l_{S_4}^{top}$	required travel distance from candidate workstation S_4 to top side along with the vertical aisle	$l_{S_4}^{top} = y_{\max} - y_{S_4}$

For each given workstation, say D_2 ($D_2 \in S_{\text{candidate}}$), to enumerate all possible paths, we further divide its opposite category C into $C1$ and $C2$ using a vertical line (as the dashed red line in Figure 5) that crosses D_2.

Let $d_{ij}^{S_1}$ denote the shortest distance of pod (i,j) to S_1, and likewise for other candidate workstations. To derive the expression of them, let

$$y_{\max} = y_{S_2},$$
$$x_{\max} = x_{S_4}. \tag{8}$$

where the positions of S_2, S_4 and other candidate workstations are shown in Figure 5. For the sake of description, we define the following distance notations in Table 2.

For the candidate workstation S_1, as discussed above,

$$d_{ij}^{S_1} = \min\{d_{ij}^{p_1}, d_{ij}^{p_2}\} = \min\left\{ l_{p,a} + l_{p,S_1}^a, l_p^{bot} + \lvert x_{ij} \rvert \right\} \tag{9}$$

Obviously, for S_2, we have

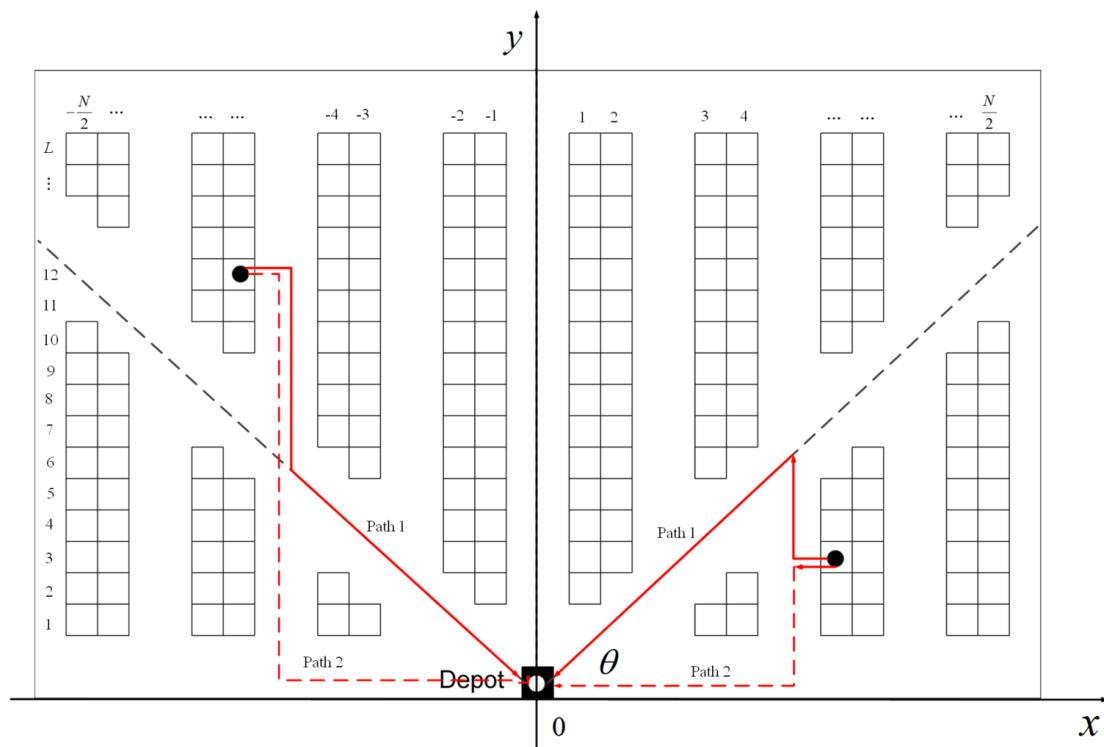$$d_{ij}^{S_2} = \lvert x_{ij} - x_{S_2} \rvert + \lvert y_{ij} - y_{S_2} \rvert \tag{10}$$

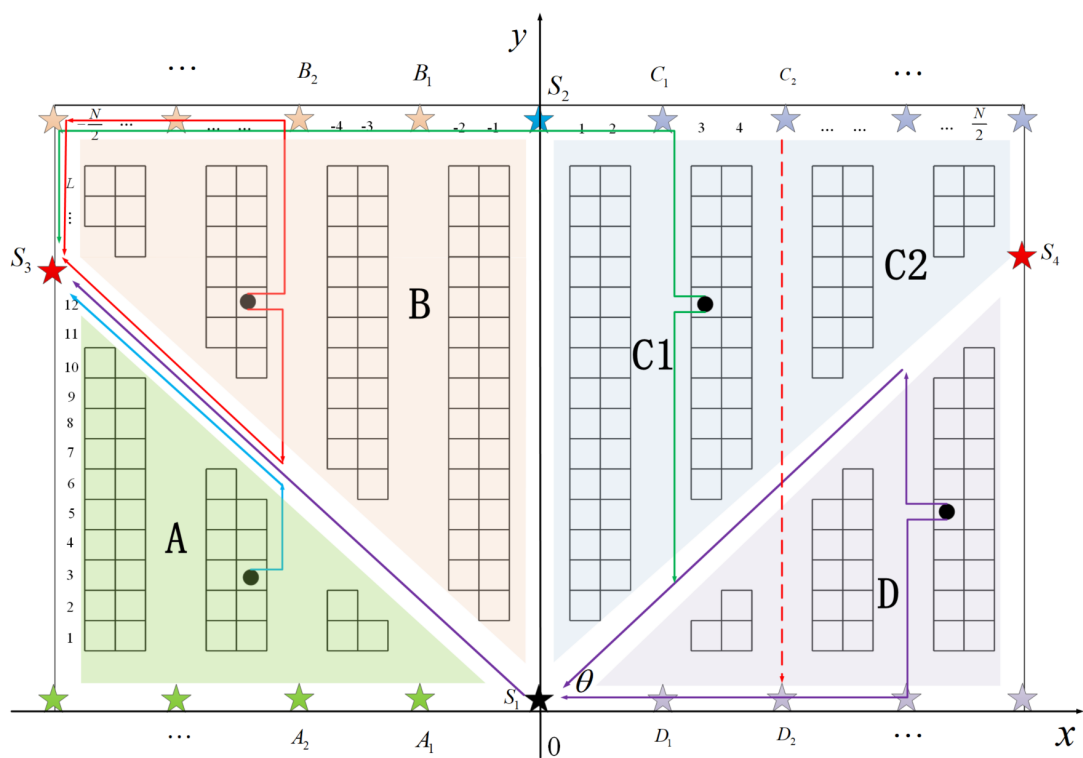**Figure 4.** Sketch of a flying-V layout with single workstation.

Figure 5. Sketch of flying-V layout with multiple workstations and four divided categories.

For S_3 (S_4 is similar to S_3), the feasible paths are shown in Figure 5. We obtain

$$d_{ij}^{S_3} = \begin{cases} l_{p,a} + l_{p,S_3}^a, & \text{if } (i,j) \in \text{area } A \\ \min\left\{l_{p,a} + l_{p,S_3}^a, \quad l_p^{top} + x_{ij} + x_{\max} + l_{S_3}^{top}\right\}, & \text{if } (i,j) \in \text{area } B \\ \min\left\{l_{p,a} + l_{p,S_1}^a + l_a, \quad l_p^{top} + x_{ij} + x_{\max} + l_{S_3}^{top}\right\}, & \text{if } (i,j) \in \text{area } C \\ \min\left\{l_{p,a} + l_{p,S_1}^a + l_a, \quad l_p^{bot} + x_{ij} + l_{S_3}^{top}\right\}, & \text{if } (i,j) \in \text{area } D \end{cases} \tag{11}$$

Similarly, we have

$$d_{ij}^{S_4} = \begin{cases} \min\left\{l_{p,a} + l_{p,S_1}^a + l_a, \quad l_p^{bot} + x_{ij} + l_{S_4}^{top}\right\}, & \text{if } (i,j) \in \text{area } A \\ \min\left\{l_{p,a} + l_{p,S_1}^a + l_a, \quad l_p^{top} + x_{ij} + x_{\max} + l_{S_4}^{top}\right\}, & \text{if } (i,j) \in \text{area } B \\ \min\left\{l_{p,a} + l_{p,S_4}^a, \quad l_p^{top} + x_{ij} + x_{\max} + l_{S_4}^{top}\right\}, & \text{if } (i,j) \in \text{area } C \\ l_{p,a} + l_{p,S_4}^a, & \text{if } (i,j) \in \text{area } D \end{cases} \tag{12}$$

For other cases,

$$d_{ij}^{A_i} = \begin{cases} l_p^{bot} + |x_{ij} - x_{A_i}|, & \text{if } (i,j) \in \text{area } A \\ l_{p,a} + l_{p,A_i}^a + l_{p,A_i}^{bot}, & \text{if } (i,j) \in \text{area } B1 \\ l_p^{bot} + |x_{ij} - x_{A_i}|, & \text{if } (i,j) \in \text{area } B2 \\ l_{p,a} + l_{p,S_1}^a + |x_{A_i}|, & \text{if } (i,j) \in \text{area } C \\ \min\left\{l_{p,a} + l_{p,S_1}^a + |x_{A_i}|, \quad l_p^{bot} + |x_{ij} - x_{A_i}|\right\}, & \text{if } (i,j) \in \text{area } D \end{cases} \tag{13}$$

$$d_{ij}^{D_i} = \begin{cases} \min\left\{l_{p,a} + l_{p,S_1}^a + |x_{D_i}|, \quad l_p^{bot} + |x_{ij} - x_{D_i}|\right\}, & \text{if } (i,j) \in \text{area } A \\ l_{p,a} + l_{p,S_1}^a + |x_{D_i}|, & \text{if } (i,j) \in \text{area } B \\ l_p^{bot} + |x_{ij} - x_{D_i}|, & \text{if } (i,j) \in \text{area } C1 \\ l_{p,a} + l_{p,D_i}^a + l_{p,D_i}^{bot}, & \text{if } (i,j) \in \text{area } C2 \\ l_p^{bot} + |x_{ij} - x_{D_i}|, & \text{if } (i,j) \in \text{area } D \end{cases} \tag{14}$$

$$d_{ij}^{B_i} = \begin{cases} l_p^{top} + |x_{ij} - x_{B_i}|, & \text{if } (i,j) \in \text{area } A1 \\ l_{p,a} + l_{p,B_i}^a + l_{p,B_i}^{a,top}, & \text{if } (i,j) \in \text{area } A2 \\ l_p^{top} + |x_{ij} - x_{B_i}|, & \text{if } (i,j) \in \text{areas } B, C \\ l_{p,a} + l_{p,D_i}^a + l_{p,D_i}^{bot}, & \text{if } (i,j) \in \text{area } C2 \\ \min\{l_p^{bot} + x_{ij} + l_{p,B_i}^a + l_{p,B_i}^{a,top}, \quad l_{p,a} + l_{p,S_1}^a + l_{S_1,B_i}^a + l_{p,B_i}^{a,top}, \\ \qquad l_p^{top} + |x_{ij} - x_{B_i}|\}, & \text{if } (i,j) \in \text{area } D \end{cases} \tag{15}$$

$$d_{ij}^{C_i} = \begin{cases} \min\{l_p^{bot} + x_{ij} + l_{p,C_i}^a + l_{p,C_i}^{a,top}, \\ \qquad l_{p,a} + l_{p,S_1}^a + l_{S_1,C_i}^a + l_{p,C_i}^{a,top}, l_p^{top} + |x_{ij} - x_{C_i}|\}, & \text{if } (i,j) \in \text{area } A \\ l_p^{top} + |x_{ij} - x_{C_i}|, & \text{if } (i,j) \in \text{areas } B, C \\ l_{p,a} + l_{p,C_i}^a + l_{p,C_i}^{a,top}, & \text{if } (i,j) \in \text{area } D1 \\ l_p^{top} + |x_{ij} - x_{C_i}|, & \text{if } (i,j) \in \text{area } D2 \end{cases} \tag{16}$$

Suppose the selected workstation set is $\mathcal{R}^*$ in the optimal layout, where $|\mathcal{R}^*| = K$, and $\mathcal{R}^* \subseteq \mathcal{R}$, then any pod $(i,j) \in \mathcal{J}$ will be assigned to the workstation with shortest

travel distance in optimal solution(this conclusion can be easily derived according to Equations (18)–(22)) , i.e., the actual travel distance of this pod in optimal solution must be

$$d_{ij}^{\text{actual}} = \min_{q \in \mathcal{R}^*} \left\{ d_{ij}^q \right\} \tag{17}$$
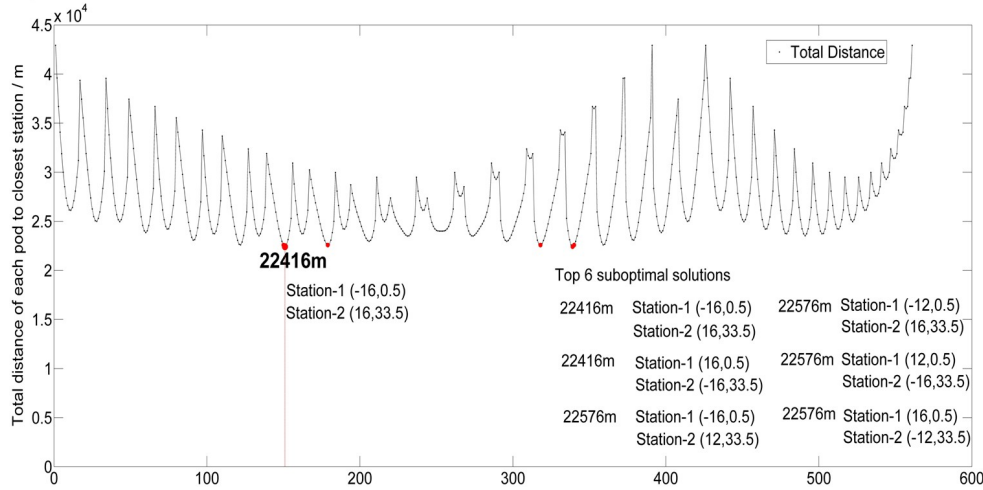
(**a**) Enumerate workstation configurations in traditional layout (561 combinations when $K = 2, N = 32, L = 30$.)
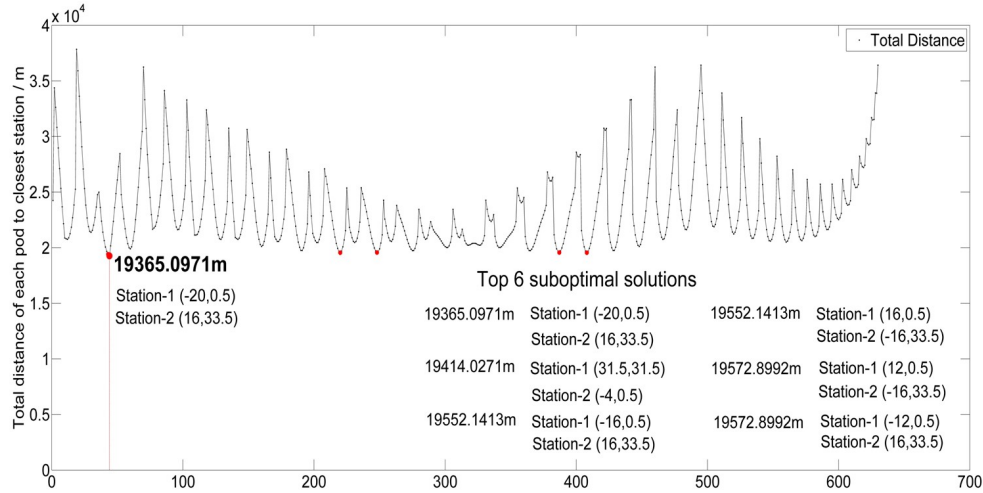
(**b**) Enumerate workstation configurations in flying-V layout (630 combinations when $K = 2, N = 32, L = 30$.)

Figure 6. Total distance for all combinations in the enumeration.

4. The Workstation Location Model

Section 3 provides the expression of the shortest travel distance from an arbitrary pod dwell location to any candidate workstation. Suppose each pod has the identical possibility of being used for goods retrieval under a random storage policy; it is tractable to solve a location problem to determine the optimal positions of multiple workstations.

One may question whether we can enumerate all the possible workstation configurations given the knowledge that number of workstations may be not that large for a single warehouse in practice. To verify this, we set up an experiment using a warehouse with width $N = 32$ and length $L = 30$, 960 pods or about 1500 m². Figure 6 shows the total distance with possible workstation combinations for both traditional layout and flying-V layout ($K = 2$). It takes about one minute of CPU time to compute the the travel distances from each pod to all candidate workstations for traditional layout and four hours for the flying-V layout. If we consider using more workstations, or a larger warehouse, the computation workload will increase sharply. For example, if we select eight workstations

in the same warehouse, there are $C_{34}^8 \approx 1.8 \times 10^7$ combinations for the traditional and $C_{36}^8 \approx 3 \times 10^7$ combinations for the flying-V layout. Therefore, the enumeration method cannot work in this case.

Based on this information, an easy-to-implement as well as efficient model should be developed to make this decision problem practical. Based on the distance expression developed above, we propose the integer programming model to minimize the total travel distance of robots. This model can determine the number and location of workstations. Please note that for the flying-V layout, we take the angle of the cross aisles as an input parameter, rather than a decision variable. The rationale behind is: (1) making it a decision variable may make the following model intractable; (2) the choices for this angle are limited owing to the discrete nature of pod locations.

Let $\mathcal{R}$ be the set of candidate workstations, $\mathcal{J}$ the set of all pod storage units (for simplicity, we call them pods), and K the number of workstations the warehouse operators decide to construct. z_r and u_{ijr} are binary variables. z_r are called *location variables*, representing whether the candidate point r is chosen as a workstation in the optimal solution or not. u_{ijr} are called *allocation variables*, indicating that whether pod (i, j) is assigned to workstation r or not. Based on the above interpretations, we redescribe the workstation location problem as a facility location problem, a typical NP-hard problem, and formulated it as an integer programming model. The proposed formulation below can be applied under both traditional and flying-V layout.

$$\min \quad \sum_{(i,j)\in\mathcal{J}} \sum_{r\in\mathcal{R}} d_{ijr} u_{ijr} \tag{18}$$

$$\sum_{r\in\mathcal{R}} z_r = K, \tag{19}$$

$$\sum_{r\in\mathcal{R}} u_{ijr} = 1, \qquad \forall (i,j) \in \mathcal{J} \tag{20}$$

$$\sum_{(i,j)\in\mathcal{J}} u_{ijr} \leqslant z_r, \qquad \forall r \in \mathcal{R} \tag{21}$$

$$z_r, u_{ijr} \in \{0, 1\}, \qquad \forall (i,j) \in \mathcal{J}, \forall r \in \mathcal{R}. \tag{22}$$

The objective function Equation (18) aims to minimize the total travel distance from any pod to its assigned workstation along the shortest path as discussed in Section 3. Constraints Equation (19) restricts that exactly K workstations are selected in optimal solution. Constraints Equation (20) specify that each pod should be assigned to exactly one workstation. Constraints Equation (21) guarantee that no pod will be assigned to candidate workstation r if it is not selected in optimal solution. Constraints Equation (22) define the variable type.

To implement the above integer programming model, we should first generate candidate workstation set $\mathcal{R}$, i.e., derive the coordinates of candidate workstations according to the interpretations in Section 3 (more specifically, Figure 3 in Section 3.1 for traditional layout and Figure 5 in Section 3.2 for flying-V layout). Then, calculate the shortest distance from each pod (i, j) to every candidate workstation, i.e., generate travel distance matrix $d_{ijr}, \forall (i, j) \in \mathcal{J}, \forall r \in \mathcal{R}$ based on Equations (9)–(16). As input parameters $\mathcal{R}, d_{ijr}$ and K are well prepared, the proposed optimization model can be successfully implemented and solved by optimization solvers, for example, CPLEX and Gurobi.

5. Numerical Experiments

We conduct a series of numerical experiments to calculate the total distance under different warehouse layout settings and explore the optimal station layouts. The main purpose of this work is to discover the patterns that identify the optimal layouts, and meanwhile, to summarize the impact of warehouse width, length and workstation number on

the robot performance. We will also compare the traditional layout with flying-V layout in terms of travel distance and space use. We use MATLAB to sketch the warehouse and index the pods and candidate workstations. Then we employ CPLEX to solve the integer programming model. All computations are executed on a PC with Intel Core i7-6700 CPU (2.8 GHz) and 16 GB RAM.

5.1. Experiment Settings

To conduct the experiments, we design instances for traditional and flying-V layouts with different parameters, including warehouse width, length, and angle of the angled cross-aisle (i.e., θ). The parameters are set as shown in Table 3. We assume the pod width as unit length, i.e., $w_p = 1$. We also set the width of the picking aisle, the top, bottom and angled cross-aisle (in flying-V layout) as 2, i.e., $w_a = 2, w_e = 2$. Typically, the storage space for a pod is 1×1 m^2 in an e-commerce warehouse. Therefore, the largest scale in our experiments is comparable to a warehouse with almost 18,000 m^2 (if $N = 48, L = 180$, then the warehouse area is $(180 + 4) \times (48 + 24 \times 2) = 17664$ m^2), which is big enough for a single warehouse in the industry. Please note that constraint Equation (20) restricts the number of workstations (K) selected in the optimal solution, and hence we can fix the workstation number easily.

Table 3. Parameter settings of the numerical experiments

		Traditional Layout Warehouses	**Flying-V Layout Warehouses**
Warehouse Size	N	6, 8, 10, 12, 14, 16, 18, 20, 24, 30,32,40,48	
	L	10, 15, 20, 25, 30, 35, 40, 50, 60, 70, 80, 90, 100,120, 150, 180	
K		1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12	
Angle of Cross-aisle		-	$\frac{5}{36}\pi, \frac{7}{36}\pi, \frac{1}{4}\pi, \frac{11}{36}\pi, \frac{13}{36}\pi$

N: number of columns in the warehouse; L: number of rows in the warehouse; K: number of workstations in the warehouse.

5.2. Computational Efficiency

The solution process mainly consists of two steps: (1) calculate the distance matrix from any pod to any candidate workstation with Equations (9)–(16); and (2) build the integer programming model, i.e., Equations (18)–(22) based on the aforementioned distance matrix and solve it. We find the computational time for step 2 is ignorable compared to step 1, therefore, we report the running time of step 1 for some scenarios in Table 4. As shown, flying-V is much more time-consuming than the traditional layout. For example, it takes 158.33 s to achieve the flying-V results while only 13.70 s for the traditional layout when the warehouse is 20 pods in width and 120 pods in length. For a warehouse 30 pods in width and 180 pods in length, the running time for flying-V is over 670 times of that for a traditional layout. This is due to an increase in path choices in the flying-V layout, but it is still acceptable as a high-level decision problem.

Table 4. Running time comparison.

N	Length	Distance Calculation Time(*s*)	
		Traditional	**Flying-V**
20	60	6.50	41.65
	80	8.90	79.17
	100	11.38	111.35
	120	13.70	158.33
30	90	21.31	473.15
	120	28.94	748.92
	150	36.41	9647.43
	180	44.84	30,384.77

5.3. Results in a Traditional Warehouse Layout

We first observe the location of workstations in the optimal solution when given different numbers of workstations and different length-to-width ratio in traditional layout. Then we will try to produce some rules of thumb to determine the workstation locations that are simple to use while without a loss of accuracy.

As part of the experimental results, the optimal solutions for instances when $N = 32$ are shown in Figure 7; more layout figures can be referenced in Appendix Figure A1. The black solid blocks represent the workstations and each square represents a pod. Using the 4-workstation case as an example, it is interesting to see that the workstations on the top and bottom sides are not symmetric; in other words, the quadrangle consisting of the four workstations is not a rectangle, but a parallelogram, which is counter-intuitive. As the number of workstations increases, the shape has the tendency to turn from a parallelogram to a rectangle.

(**a**) $L = 40, N = 32, K = 3, 4, 5, 6$

(**b**) $L = 50, N = 32, K = 3, 4, 5, 6$

(**c**) $L = 50, N = 32, K = 3, 4, 5, 6$
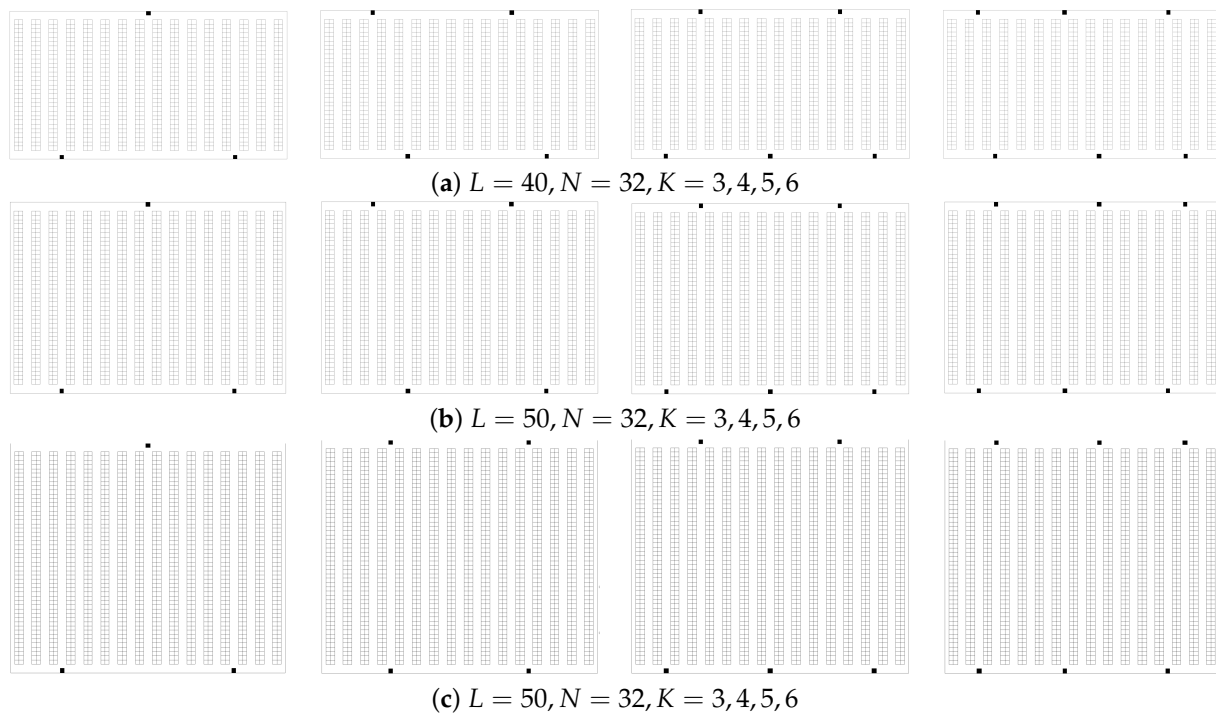
Figure 7. Optimal workstation layout under traditional layout.

We keep the warehouse length as 40 while varying the width. The optimal workstation layout is illustrated in Figure 8. As the width of the warehouse increases, the unsymmetrical

nature of the workstations becomes more obvious; that is, the workstation layout changes from that of a rectangle to a parallelogram.
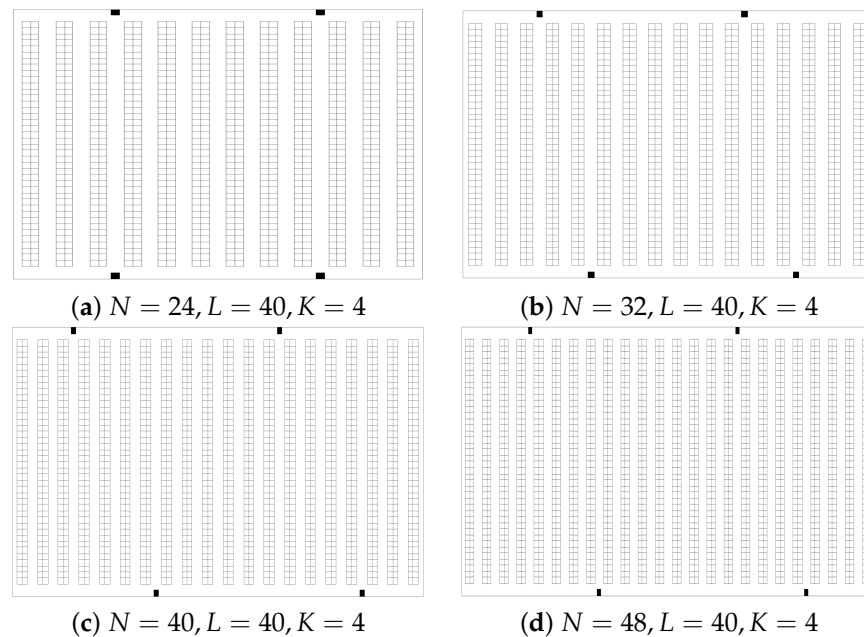

(**a**) $N = 24, L = 40, K = 4$

(**b**) $N = 32, L = 40, K = 4$

(**c**) $N = 40, L = 40, K = 4$

(**d**) $N = 48, L = 40, K = 4$

Figure 8. Optimal workstation layout under different N (i.e., number of columns).

As it would be time-consuming to calculate the distance matrix and acquire the optimal solution by solving the proposed workstation location model (i.e., Equations (18)–(22), which is NP-hard) for medium and large-scale warehouses, even though it is tractable, we herein propose some general design rules by which the workstations can be easily deployed. We perform numerical experiments to verify the validity of these rules. If the number of workstations to be located on the top (or the bottom) is n, the width (in meters) of the warehouse is w, then we have two rules:

- **rule 1**: $2n$ rule. By this rule, the edge of the warehouse is equally divided by points with interval of $\frac{w}{2n}$ and workstations are inserted at the dividing points with odd index.
- **rule 2**: $n + 1$ rule. By this rule, workstations are inserted at the dividing points with equal interval of $\frac{w}{n+1}$.

Figure 9 demonstrates examples of rules 1 and 2 with different numbers of workstations, and compares them with the optimal workstation locations.

To evaluate the proposed two rules, we conduct two set of experiments, with an odd and even number of workstations, respectively, to compare the travel distance obtained by the proposed optimization model, the $2n$ rule and the $n + 1$ rule. The results are reported in Tables 5 and 6, in which the GAP represent the percentage of extra travel distance compared with the optimal solution. Both tables show that the two rules are effective, because even the maximum Gap is around 5%. Between the two rules, the $2n$ rule exhibits better performance with a Gap less than 1.5%. Please note that if the resultant workstations are not located at the end of a picking aisle according to the rules, we will adjust them to a location nearby.
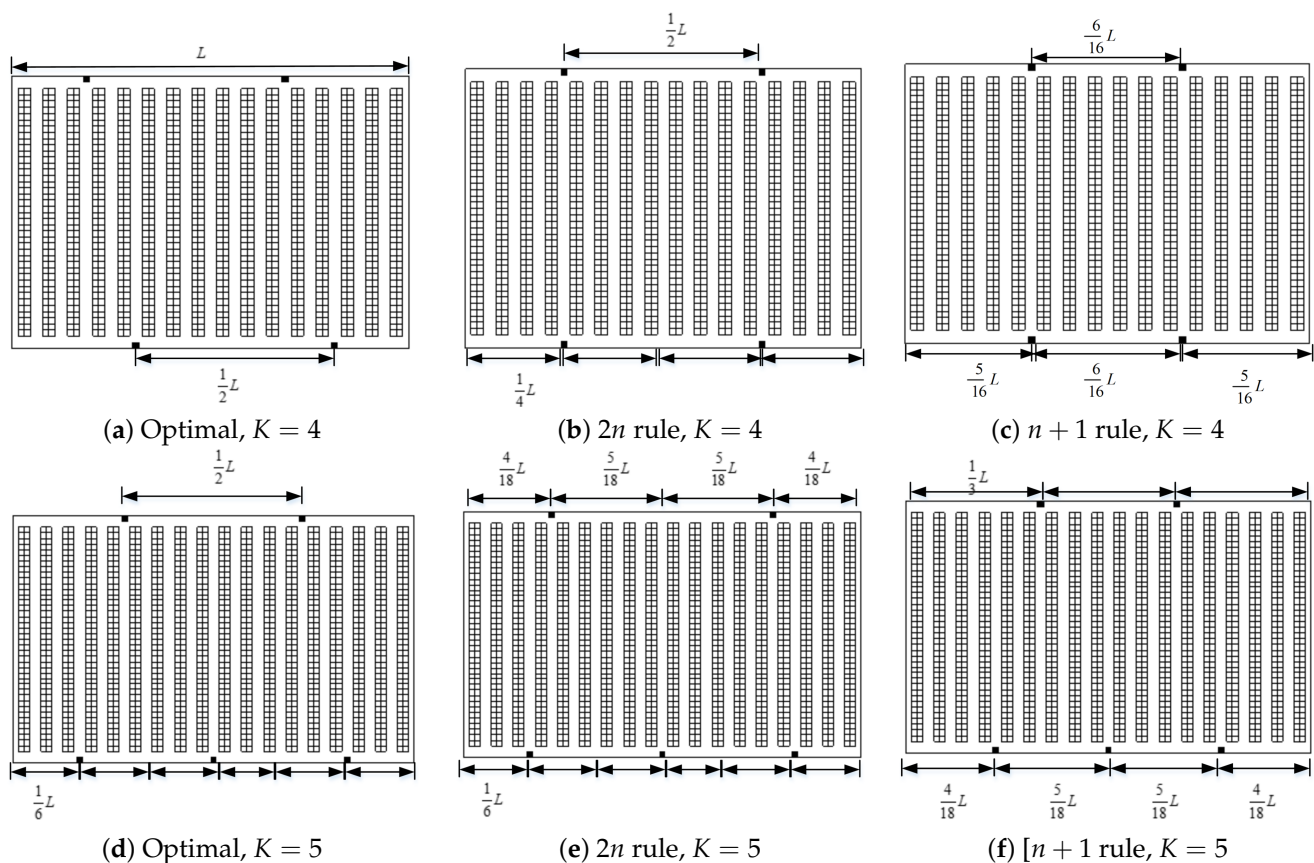
Figure 9. Workstation layout by optimization model and two simple rules ($N = 32, L = 40$).

Table 5. Evaluation of two rules for traditional layout with an odd number of workstations

N	L	$K = 3$					$K = 5$				
		TD_{op}^a	TD_{2n}^b	gap(%)	TD_{n+1}^c	gap(%)	TD_{op}	TD_{2n}	gap(%)	TD_{n+1}	gap(%)
32	30	18,560	18,816	1.38	19,552	5.34	14,720	14,720	0.00	14,986	1.81
	40	28,560	28,736	0.62	29,552	3.47	22,960	22,960	0.00	23,296	1.46
	50	40,160	40,256	0.24	41,152	2.47	32,800	32,800	0.00	33,206	1.24
	60	53,346	53,376	0.06	54,352	1.89	44,240	44,240	0.00	44,716	1.08
	70	68,096	68,096	0.00	69,152	1.55	57,280	57,280	0.00	57,826	0.95
	80	84,416	84,416	0.00	85,552	1.35	71,920	71,920	0.00	72,536	0.86

a: optimal travel distance obtained by the proposed integer programming; *b*: travel distance obtained by $2n$ rule; and *c*: travel distance obtained by $n + 1$ rule.

Table 6. Evaluation of two rules with an even number of workstations

N	L	$K = 4$					$K = 6$				
		TD_{op}^a	TD_{2n}^b	gap(%)	TD_{n+1}^c	gap(%)	TD_{op}	TD_{2n}	gap(%)	TD_{n+1}	gap(%)
32	30	16,192	16,320	0.79	16800	3.75	13,734	13,920	1.35	14,400	4.85
	40	24,912	24,960	0.19	25,600	2.76	21,564	21,760	0.91	22,400	3.88
	50	35,200	35,200	0.00	36,000	2.27	30,994	31,200	0.66	32,000	3.25
	60	47,040	47,040	0.00	48,000	2.04	42,024	42,240	0.51	43,200	2.80
	70	60,480	60,480	0.00	61,600	1.85	54,654	54,880	0.41	56,000	2.46
	80	75,520	75,520	0.00	76,800	1.69	68,884	69,120	0.34	70,400	2.20

5.4. Results on Flying-V Warehouse Layout

5.4.1. The Optimal Layout of Flying-V

　　When setting one workstation in the warehouse, it is always located on the intersection of a cross-aisle on the bottom edge. For multiple workstations, the candidates may be located on: (1) the top and bottom edges, (2) the intersection of two angled cross aisles, or (3) the intersection of oblique aisles and left and right edges, depending on the cross-aisle angle. Figure 10 illustrates the optimal workstation locations with different quantities and different warehouse lengths; more can be referenced in Appendix A Figures A2–A4. The workstations seem to disperse differently in the warehouse depending on the length-to-width ratio. It is surprising to see that for three workstations, the optimal positions do not include any aisle intersections. For four workstations, the three cross-aisle intersections are only presented in Figure 10c. It can be concluded that when angled cross aisles reach the upper corner of the warehouse or close to it, the endpoints of the cross-aisle are probably not set as workstations.
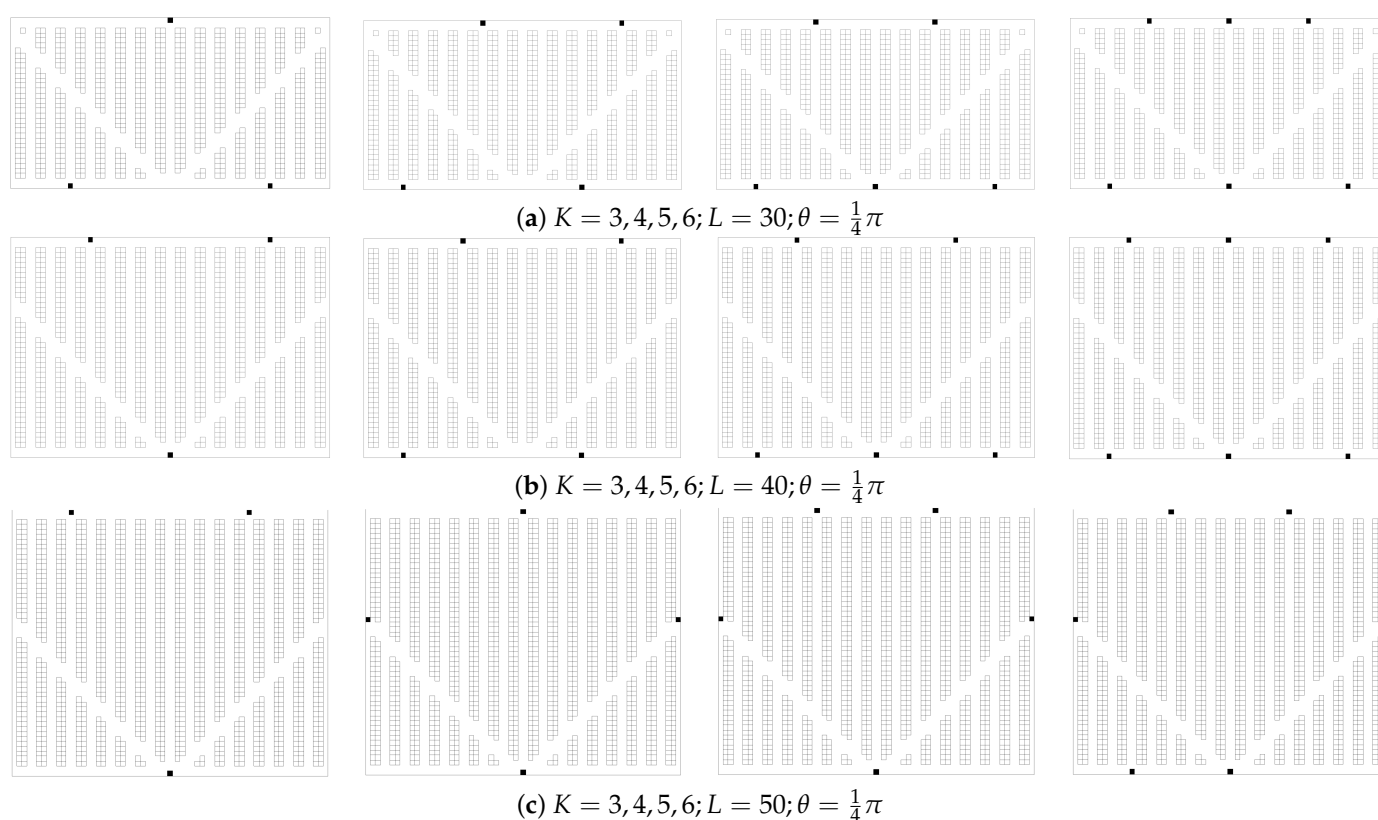

(**a**) $K = 3, 4, 5, 6; L = 30; \theta = \frac{1}{4}\pi$

(**b**) $K = 3, 4, 5, 6; L = 40; \theta = \frac{1}{4}\pi$

(**c**) $K = 3, 4, 5, 6; L = 50; \theta = \frac{1}{4}\pi$

Figure 10. Optimal flying-V warehouse layout with different number of rows.

　　We also apply the 2n rule and $n + 1$ rule to the flying-V layout to evaluate its effectiveness. To apply the rules, we first fix the cross-aisle intersections with any edges as workstations, and then we locate the remaining workstations using the rules. Tables 7 and 8 show the comparative results obtained by optimization model, 2n rule and $n + 1$ rule, by varying the warehouse length. N is set as 16, and θ is $\frac{1}{4}\pi$. In Tables 9 and 10 the results are derived by changing the cross-aisle angle and fixing L as 80. It seems that the GAP is not as small as that of traditional layout. For example, when the warehouse length is 30 (here the length-to-width ratio is roughly 1.0), the GAP for $n + 1$ rule can reach as large as 15.39%. However, as the length (or, the length-to-width ratio) increases, the GAP for both 2n rule and $n + 1$ rule decreases dramatically to an acceptable level. Among them, unlike in traditional layout situations, the $n + 1$ rule seems better than the 2n rule that the GAP is normally less than 5%, except when the angle is as much as $\frac{13}{36}\pi$.

Table 7. Evaluation of two rules for a flying-V layout with varying lengths and odd workstation numbers.

L	K = 3					K = 5				
	TD_{op}^a	TD_{2n}^b	gap(%)	TD_{n+1}^c	gap(%)	TD_{op}	TD_{2n}	gap(%)	TD_{n+1}	gap(%)
30	15,817.00	16,231.29	2.62	17,014.56	7.57	12,341.79	12,405.33	0.51	14,198.08	15.04
40	25,211.04	25,239.64	0.11	26,032.61	3.26	19,942.51	19,942.51	0.00	20,725.88	3.93
50	35,804.80	35,816.23	0.03	36,453.01	1.81	28,742.93	29,407.60	2.31	28,955.16	0.74
60	47,627.74	47,627.74	0.00	48,242.22	1.29	38,656.56	40,590.70	5.00	38,788.78	0.34
70	60,928.13	60,928.13	0.00	61,622.61	1.14	50,167.67	53,375.80	6.39	50,222.41	0.11
80	75,828.51	75,828.51	0.00	76,602.99	1.02	63,236.30	67,760.89	7.16	63,256.03	0.03

Table 8. Evaluation of two rules for a flying-V layout with varying lengths and odd workstation numbers.

L	K = 4					K = 6				
	TD_{op}^a	TD_{2n}^b	gap(%)	TD_{n+1}^c	gap(%)	TD_{op}	TD_{2n}	gap(%)	TD_{n+1}	gap(%)
30	13,930.99	14,281.08	2.51	15,106.58	8.44	11,647.16	11,807.75	1.38	13,439.90	15.39
40	22,452.60	22,542.24	0.40	22,562.43	0.49	18,725.86	18,922.95	1.05	19,647.70	4.92
50	31,848.53	32,581.00	2.30	31,848.53	0.00	27,351.13	27,828.04	1.74	27,556.98	0.75
60	42,905.27	44,448.00	3.60	42,905.27	0.00	37,070.60	38,531.14	3.94	37,070.60	0.00
70	55,470.00	58,023.00	4.60	55,618.90	0.27	48,164.98	50,916.24	5.71	48,184.23	0.04
80	69,477.01	73,148.87	5.28	69,932.52	0.66	60,795.30	64,901.33	6.75	60,897.85	0.17

Table 9. Evaluation of two rules for a flying-V layout with varying angles and odd workstation numbers.

θ	K = 3					K = 5				
	TD_{op}^a	TD_{2n}^b	gap(%)	TD_{n+1}^c	gap(%)	TD_{op}	TD_{2n}	gap(%)	TD_{n+1}	gap(%)
$\frac{5}{36}\pi$	76,901.71	76,901.71	0.00	77,816.04	1.19	66,284.88	69,358.11	4.64	66,616.63	0.50
$\frac{7}{36}\pi$	75,857.36	75,857.36	0.00	76,698.2	1.11	64,281.48	68,445.78	6.48	64,477.75	0.31
$\frac{9}{36}\pi$	75,828.51	75,828.512	0.00	76,602.99	1.02	63,236.3	67,760.0	7.16	63,256.03	0.03
$\frac{11}{36}\pi$	77,297.00	77,297.00	0.00	78,009.14	0.92	63,743.06	67,110.24	5.28	63,743.06	0.00
$\frac{13}{36}\pi$	75,768.65	75,768.65	0.00	76,679.28	1.20	63,705.95	63,705.95	0.00	66,339.35	4.13

Table 10. Evaluation of two rules for a flying-V layout with varying angles and even workstation numbers.

θ	K = 4					K = 6				
	TD_{op}^a	TD_{2n}^b	gap(%)	TD_{n+1}^c	gap(%)	TD_{op}	TD_{2n}	gap(%)	TD_{n+1}	gap(%)
$\frac{5}{36}\pi$	71,593.30	74,140.09	3.56	74,373.93	3.88	63,549.64	66,542.36	4.71	65,326.33	2.80
$\frac{7}{36}\pi$	70,067.83	73,469.44	4.85	71,725.79	2.37	61,665.91	65,606.88	6.39	63,191.80	2.47
$\frac{9}{36}\pi$	69,477.01	73,148.87	5.28	69,932.52	0.66	60,795.30	64,901.33	6.75	60,897.85	0.17
$\frac{11}{36}\pi$	69,594.93	72,503.00	4.18	69,594.93	0.00	61,283.73	64,252.63	4.84	62,465.60	1.93
$\frac{13}{36}\pi$	68,592.55	68,668.83	0.11	70,926.76	3.40	60,709.21	61,344.47	1.05	65,064.70	7.17

5.4.2. The Influence of the Cross-Aisle Angle on Total Travel Distance

When there is only one workstation in the warehouse with a flying-V layout, the total distance will decrease as the cross-aisle angle increases, until the cross aisles gradually close

at the top corners. This observation is in line with the conclusion found by Cardona et al. (2012) [24] using a fishbone layout.

We undertake a series of experiments to explore the features of the "optimal" cross-aisle angle for a variety of warehouse configurations with multiple workstations, as shown in Table 11. It seems that a larger angle is preferable in most cases. Meanwhile, the cross-aisle angle has very slight impact on the space use. Therefore, we suggest that the "optimal" cross-aisle angle should be set as large as possible as long as the angled cross aisles intersect with the left or right edge of the warehouse.

Table 11. The influence of aisle angle on distance and space use

θ	$K = 3$	$K = 4$	$K = 5$	$K = 6$	$K = 7$	$K = 8$	Space Use (%)
			$N = 32, L = 80$				
$\frac{5}{36}\pi$	76,901.7	71,593.3	66,284.9	63,549.6	61,934.3	60,824.3	45.28
$\frac{7}{36}\pi$	75,857.4	70,067.8	64,281.5	61,665.9	60,147.6	59,027.0	45.24
$\frac{9}{36}\pi$	75,828.51	69,477.0	63,236.3	60,795.3	59,303.6	57,811.8	45.42
$\frac{11}{36}\pi$	77,297.00	69,594.9	63,743.1	61,283.7	58,824.4	56,682.0	45.57
$\frac{13}{36}\pi$	75,768.65	68,592.6	63,706.0	60,709.2	57,779.5	56,033.5	43.60
			$N = 40, L = 120$				
$\frac{5}{36}\pi$	204,151.1	191,361.1	178,571.2	171,840.2	168,269.2	165,519.2	46.75
$\frac{7}{36}\pi$	201,360.3	187,379.6	173,437.2	166,905.9	163,516.0	160,815.9	46.75
$\frac{9}{36}\pi$	200,535.1	185,138.6	169,937.4	163,692.0	160,468.0	157,244.0	46.88
$\frac{11}{36}\pi$	202,678.5	184,440.6	169,065.8	163,464.1	158,475.0	153,485.8	46.98
$\frac{13}{36}\pi$	202,169.7	185,125.5	172,327.0	163,726.9	155,126.9	150,602.5	45.67

5.5. Comparison of Traditional and Flying-V Layouts

In this section, we will identify the advantages of flying-V layout for RMFS applications. Both travel distance and space use will be investigated.

5.5.1. Total Travel Distance

We compare the traditional and flying-V layout in terms of total travel distance by varying number of workstations, warehouse length and width, as in Tables 12 and 13 where Diff. (%) is the percentage of travel distance saving compared to traditional layout. Both tables show that the flying-V layout outperforms the traditional layout in terms of travel distance, and the saving ratio depends on the warehouse size and workstation number. To be specific, in Table 12, about 9~26% distance saving can be observed with a length-to-width ranging from 0.6 to 1.6. When given the number of picking aisles, the advantage of distance saving dwindles as the length increases. In Table 13, around 8~18% distance saving can be observed when the length-to-width ratio ranges from 0.8 to 1.33.

Table 12. Total distance comparison ($N = 20$).

K	L	$\frac{L}{W}$ *	Travel Distance Traditional	Flying-V	Reduction	Diff.(%)
1	20	0.6	8600.00	6321.85	2278.15	26.49
	40	1.1	25,200.00	20,778.70	4421.30	17.54
	60	1.6	49,800.00	43,235.56	6564.44	13.18
2	20	0.6	6312.00	5175.76	1136.24	18.00
	40	1.1	17,200.00	14724.78	2475.22	14.39
	60	1.6	31,800.00	28253.21	3546.79	11.15
3	20	0.6	5224.00	4201.00	1023.00	19.58
	40	1.1	14,864.00	12,887.36	1976.64	13.30
	60	1.6	28,504.00	25,455.79	3048.21	10.69
4	20	0.6	4600.00	3700.90	899.10	19.55
	40	1.1	13260.00	11732.31	1527.69	11.52
	60	1.6	25920.00	23562.41	2357.59	9.10

$\frac{L}{W}$ represents length-to-width ratio.

Table 13. Total distance comparison ($L = 60$).

K	L	$\frac{L}{W}$ *	Travel Distance Traditional	Flying-V	Reduction	Diff.(%)
1	24	1.33	62,640.00	53,426.30	9213.70	14.71
	32	1.0	91,200.00	75,871.32	15,328.68	16.81
	40	0.8	123,600.00	101,495.84	22,104.16	17.88
2	24	1.33	41,040.00	36177.13	4862.87	11.85
	32	1.0	62,400.00	54731.05	7668.95	12.29
	40	0.8	87,312.00	77,246.66	10,065.34	11.53
3	24	1.33	36160.00	32,093.01	4066.99	11.25
	32	1.0	53,346.00	47,627.74	5718.26	10.72
	40	0.8	72,656.00	65,730.83	6925.17	9.53
4	24	1.33	32,400.00	29,481.72	2918.28	9.01
	32	1.0	47,040.00	42,905.27	4134.73	8.79
	40	0.8	63,584.00	58,440.49	5143.51	8.09
5	32	1.0	44,240.00	38,656.56	5583.44	12.62
	40	0.8	59,036.00	52,780.39	6255.61	10.60
6	32	1.0	42024.00	37070.60	4953.40	11.79
	40	0.8	55,644.00	49,890.49	5753.51	10.34
7	32	1.0	40604.00	35672.87	4931.13	12.14
	40	0.8	53,532.00	47,000.59	6531.41	12.20
8	32	1.0	39360.00	34,275.14	5084.86	12.92
	40	0.8	51,750.00	44,912.53	6837.47	13.21

The tremendous distance saving shown in the flying-V layout can be explained by the inserted crossing aisles, as they enable diagonal paths rather than orthogonal ones. When the warehouse length increases, the vertical path dominates the diagonal path, weakening the advantage of distance saving. Likewise, more workstations may also eliminate distance saving, as presented in the tables.

5.5.2. Space Use

As additional cross aisles are inserted in a flying-V warehouse, space use is sacrificed. Therefore, how to tradeoff between the benefit of travel saving and the disadvantage of space waste is a practical problem. To address this, we integrate in Table 14 experimental results of the two aspects together. When the warehouse is relatively small, as much as 18.75% travel distance can be saved but the space use rate decreases 6.94%. However, as the

warehouse size increases, a roughly 10% travel saving can be obtained for just 2% or 3% space loss. This suggests that flying-V is applicable if space is not a bottleneck resource, or if the warehouse is of a medium or large size, for example, over 5000 m^2 (this is equivalent to $N = 24, L = 60$).

Table 14. Space use on different layouts.

N	L	Travel Distance		Diff.(%)	Space Use (%)		Diff.(%)
		Traditional	Flying-V		Traditional	Flying-V	
	20	5920.00	4810.00	18.75	41.67%	34.72%	6.94
	30	10,800.00	9411.01	12.86	44.12	38.85	5.27
24 ($K = 4$)	40	16,800.00	14,894.50	11.34	45.45	41.38	4.07
	50	24,000.00	21,654.63	9.77	46.30	42.98	3.32
	60	32,400.00	29,481.72	9.01	46.88	44.08	2.80
	60	86,214.00	78,728.02	8.68	46.88%	43.88%	2.99
	90	170,138.00	150,645.72	11.46	47.87	45.83	2.04
30 ($K = 8$)	120	280,988.00	249,396.93	11.24	48.39	46.84	1.55
	150	418,838.00	374,125.60	10.68	48.70	47.46	1.24
	180	583,688.00	525,673.84	9.94	48.91	47.87	1.04

6. Conclusions

This work addresses the warehouse layout design problem in the context of a RMFS, which is an emerging approach that is attracting a lot of attention in the e-commerce industries. Although there exist extensive research in this field, most studies focus on warehouses with a traditional layout. To the best of our knowledge, this might be the first attempt to consider a non-traditional warehouse layout for a RMFS. To improve the efficiency of the system at a high level, this paper optimizes the location of workstations from a strategic design perspective. Aisle configuration is also taken into consideration to investigate the effects on expected travel distance. We first deduce the expression of travel distance from any storage location to any candidate workstation; then, we propose an integer programming model to determine the picking station location, which is a typical facility location problem. Based on this methodology, we design and conduct a variety of numerical experiments on both traditional and flying-V layouts, through which we find two rules of thumb that can replace the complex computation process: $2n$ rule and $n + 1$ rule. By using theoretical and experimental studies, we draw the following conclusions:

1. The flying-V layout can save 8~26% of robot travel distance compared with a traditional layout, and the distance saving is related to the number of workstations and warehouse size. When the warehouse of a medium or large size, saving 10% in distance may only necessitate sacrificing about 2~3% of space use.
2. For a warehouse with a traditional layout, we can simply use the $2n$ rule to locate workstations; likewise, for the flying-V layout, the n+1 rule can be used to approximately locate the workstations. The travel distance gap for the two cases would be less than roughly 1.5% and 5%, respectively.
3. The "optimal" cross-aisle angle can be set as large as possible as long as the angled cross-aisle intersects the left or right edge of the warehouse.

There exist several limitations in this study. First, this work is based on an assumption of random storage policy. However, as in some other existing RMFS related research, more storage policies, especially the class-based policy, are worthy of further investigation. Second, the proposed IP model is solved by solver rather than well-designed efficient algorithms. Therefore, developing advanced solution methodologies for this problem and compare their performance with existing facility location related algorithms maybe a future research direction. Moreover, the optimum number of workstations is not determined in this work. We conduct the sensitivity analysis over different number of workstations, whereas the tradeoff between distance saving and construction cost is not covered. Another limitation is due to the efficiency evaluated criteria. Rather than travel distance, future

studies can consider travel time or the makespan. In addition, more non-traditional warehouse layout designs can be explored in the future, such as fishbone or chevron ones. Finally, to make this work more realistic, the impact of batching decisions on layout design can also be taken into consideration in the future research, and extensive sensitivity analysis over different batching time strategies can be explored as well.

Author Contributions: Conceptualization, X.L. and L.F.; methodology, X.L. and L.F.; software, X.Y. and J.Z.; formal analysis, X.Y. and M.Q.; investigation, J.Z.; writing–original draft preparation, M.Q., X.L., L.F. and X.Y.; writing–review and editing, X.Y. and X.L.; visualization, J.Z., X.L. and X.Y.; supervision, M.Q.; funding acquisition, M.Q. All authors have read and agreed to the published version of the manuscript.

Data Availability Statement: Not applicable.

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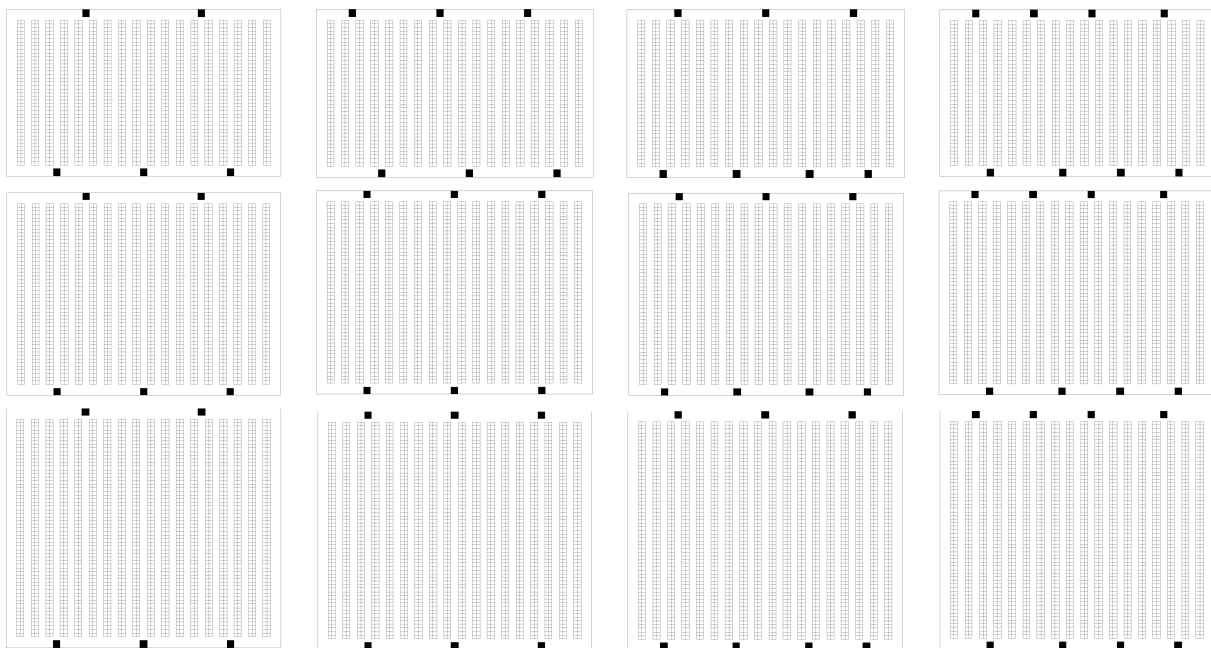
Appendix A

Figure A1. Optimal workstation layout in traditional warehouses ($N = 36$; $L = 40, 50, 60$; $K = 5, 6, 7, 8$).
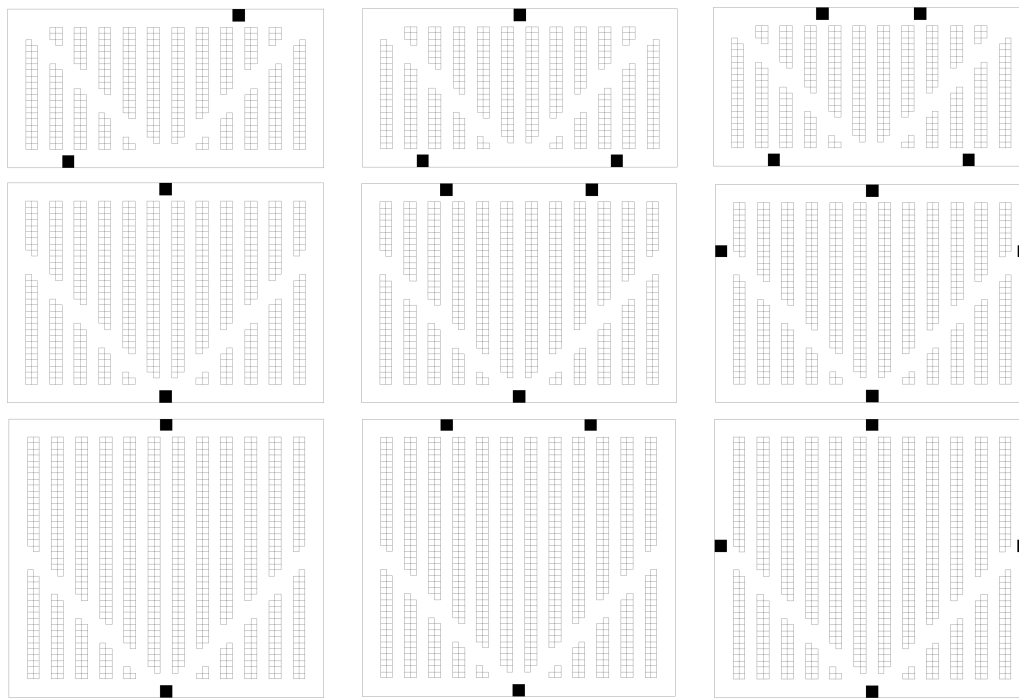
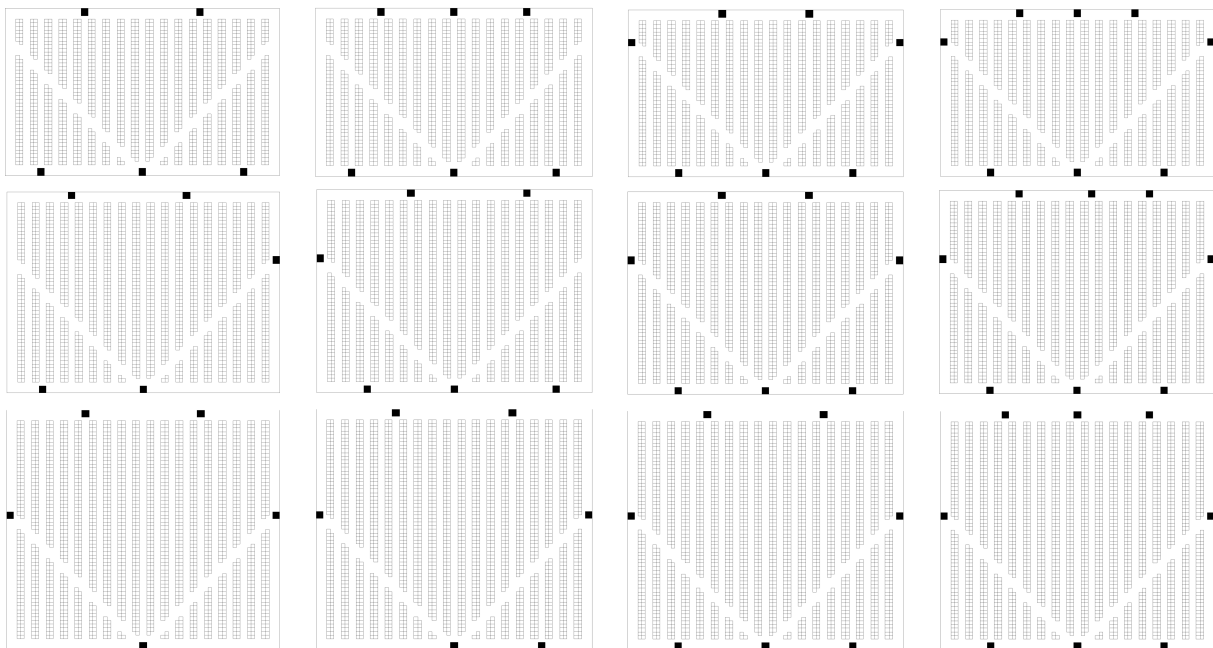**Figure A2.** Optimal workstation layout in flying-V warehouses ($N = 24; L = 20, 30, 40; K = 2, 3, 4$).

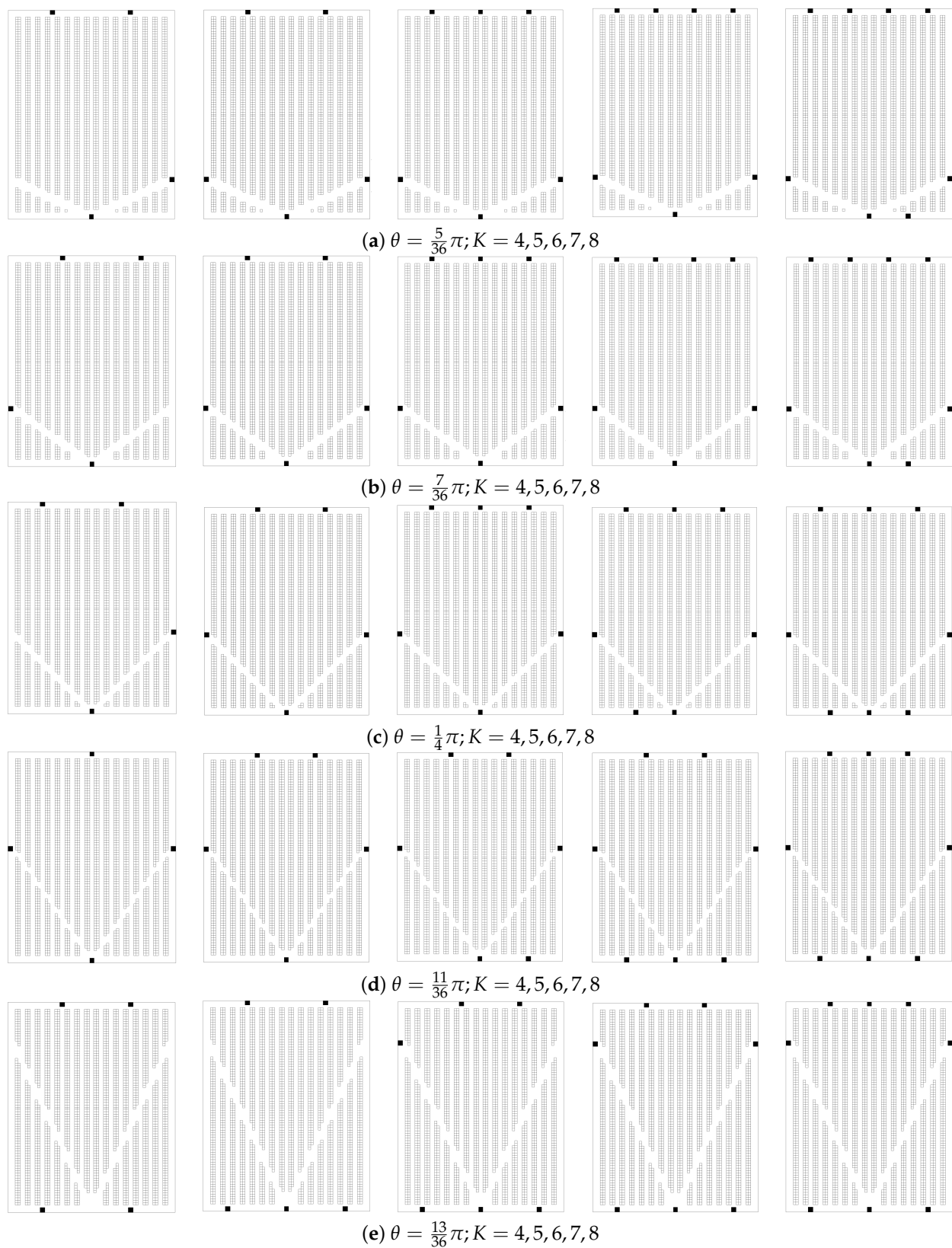**Figure A3.** Optimal workstation layout in flying-V warehouses ($N = 36; L = 40, 50, 60; K = 5, 6, 7, 8$).

(a) $\theta = \frac{5}{36}\pi$; $K = 4, 5, 6, 7, 8$

(b) $\theta = \frac{7}{36}\pi$; $K = 4, 5, 6, 7, 8$

(c) $\theta = \frac{1}{4}\pi$; $K = 4, 5, 6, 7, 8$

(d) $\theta = \frac{11}{36}\pi$; $K = 4, 5, 6, 7, 8$

(e) $\theta = \frac{13}{36}\pi$; $K = 4, 5, 6, 7, 8$

Figure A4. Optimal workstation layout in flying-V warehouses with various angles ($N = 32$; $L = 80$).

References

1. Gardrat, M. Urban growth and freight transport: From sprawl to distension. *J. Transp. Geogr.* **2021**, *91*, 102979.
2. SteadieSeifi, M.; Dellaert, N.P.; Nuijten, W.; Woensel, T.V.; Raoufi, R. Multimodal freight transportation planning: A literature review. *Eur. J. Oper. Res.* **2014**, *233*, 1–15.

3.   Abioye, O.F.; Dulebenets, M.A.; Kavoosi, M.; Pasha, J.; Theophilus, O. Vessel schedule recovery in liner shipping: Modeling alternative recovery options. *IEEE Trans. Intell. Transp. Syst.* **2020**, doi:10.1109/TITS.2020.2992120.

4.   Veerappan, M.; Sahu, P.K.; Pani, A.; Patil, G.R.; Sarkar, A.K. Analysing and modelling the relationship between air freight movement and airport characteristics in India. *Transp. Res. Procedia* **2020**, *48*, 74–92.

5.   Boysen, N.; Koster, R.; Weidinger, F. Warehousing in the e-commerce era: A survey. *Eur. J. Oper. Res.* **2021**, *277*, 396–411.

6.   Dulebenets, M.A. A Delayed Start Parallel Evolutionary Algorithm for just-in-time truck scheduling at a cross-docking facility. *Int. J. Prod. Econ.* **2019**, *212*, 236–258.

7.   Kumar, A.; Anbanandam, R. Assessment of environmental and social sustainability performance of the freight transportation industry: An index-based approach. *Transp. Policy* **2020**, doi:10.1016/j.tranpol.2020.01.006.

8.   De Koster, R.; Le-Duc, T.; Roodbergen, K.J. Design and control of warehouse order picking: A literature review. *Eur. J. Oper. Res.* **2007**, *182*, 481–501.

9.   Mountz, M. Kiva the Disrupter. *Harv. Bus. Rev.* **2012**, *90*, 74–80.

10.   Lamballais, T.; Roy, D.; De Koster, M.B.M. Estimating performance in a Robotic Mobile Fulfillment System. *Eur. J. Oper. Res.* **2017**, *256*, 976–990.

11.   Boysen, N.; Briskorn, D.; Emde, S. Parts-to-picker based order processing in a rack-moving mobile robots environment. *Eur. J. Oper. Res.* **2017**, *262*, 550–562.

12.   Yu, M. The impact of order batching and picking area zoning on order picking system performance. *Eur. J. Oper. Res.* **2009**, *198*, 480–490.

13.   Barros, Í.R.C.; Nascimento, T.P. Robotic Mobile Fulfillment Systems: A survey on recent developments and research opportunities. *Robot. Auton. Syst.* **2021**, *137*, 103729.

14.   Dukic, G.; Opetuk, T. *Warehousing in the Global Supply Chain: Advanced Models, Tools and Applications for Storage Systems*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 55–69.

15.   Roodbergena, K.J.; Vis, I.F.A.; Taylor, G.D. Simultaneous determination of warehouse layout and control policies. *Int. J. Prod. Res.* **2015**, *53*, 3306–3326.

16.   Gue, K.R.; Meller, R.D. Aisle configurations for unit-load warehouses. *IIE Trans.* **2009**, , 171–182.

17.   White, J. Optimum Design of Warehouses Having Radial Aisles. *IIE Trans.* **1972**, *4*, 333–336.

18.   Gue, K.R.; Ivanović, G.; Meller, R.D. A unit-load warehouse with multiple pickup and deposit points and non-traditional aisles. *Transp. Res. Part Logist. Transp. Rev.* **2012**, *48*, 795–806.

19.   Hu, H.; Lee, B.K.; Huang, Y.; Lee, L.H.; Chew, E.P. Performance analysis on transfer platforms in frame bridge based automated container terminals. *Math. Probl. Eng.* **2013**, *2013*, 593847.

20.   Iris, Ç.; Christensen, J.; Pacino, D.; Ropke, S. Flexible ship loading problem with transfer vehicle assignment and scheduling. *Transp. Res. Part B Methodol.* **2018**, *111*, 113–134.

21.   Boysen, N.; Koster, R.; Füßler, D. The forgotten sons: Warehousing systems for brick-and-mortar retail chains. *Eur. J. Oper. Res.* **2021**, *288*, 361–381.

22.   Wu, S.; Chi, C.; Wang, W.; Wu, Y. Research of the layout optimization in robotic mobile fulfillment systems. *Int. J. Adv. Robot. Syst.* **2020**, *17*, doi:10.1177/1729881420978543.

23.   Bortolini, M.; Faccio, M.; Gamberi, M.; Manzini, R. Diagonal cross-aisles in unit load warehouses to increase handling performance. *Int. J. Prod. Econ.* **2015**, *170*, 838–849.

24.   Cardona, L.F.; Rivera, L.; Martínez, H.J. Analytical study of the Fishbone Warehouse layout. *Int. J. Logist. Res. Appl.* **2012**, *15*, 365–388.

25.   Öztürkoğlu, Ö.; Gue, K.R.; Meller, R.D. Optimal unit-load warehouse designs for single-command operations. *IIE Trans.* **2012**, *44*, 459–475.

26.   Venkitasubramony, R.; Adil, G.K. Analytical models for pick distances in fishbone warehouse based on exact distance contour. *Int. J. Prod. Res.* **2016**, *54*, 1–22.

27.   Merschformann, M.; Lamballais, T.; De Koster, M.B.M.; Suhl, L. Decision Rules for Robotic Mobile Fulfillment Systems. *Oper. Res. Perspect.* **2019**, *6*, 100–128.

28.   Mesa, A.; Masel, D.; De Koster, M.B.M.; Suhl, L. A Non-Traditional Warehouse Layout with Multiple Pickup and Deposit Points. In Proceedings of the 2017 Industrial and Systems Engineering Conference, Pittsburgh, PA, USA, 20–23 May 2017; pp.163–168.

29.   Öztürkoğlu, Ö.; Gue, K.R.; Meller, R.D. A constructive aisle design model for unit-load warehouses with multiple pickup and deposit points. *Eur. J. Oper. Res.* **2014**, *236*, 382–394.

30.   Yuan, R.; Cezik, T.; Graves, S.C. Velocity-Based Storage Assignment in Semi-Automated Storage Systems. *Prod. Oper. Manag.* **2019**, *28*, 354–373.

31.   Zou, B.; Gong, Y.; Xu, X.; Yuan, Z. Assignment rules in robotic mobile fulfilment systems for online retailers. *Int. J. Prod. Res.* **2017**, *55*, 6175–6192.

32.   Caron, F.; Marchet, G.; Perego, A. Optimal layout in low-level picker-to-part systems. *Int. J. Prod. Res.* **2000**, *38*, 101–117.

33.   Pohl, L.M.; Meller, R.D.; Gue, K.R. An analysis of dual-command operations in common warehouse designs. *Transp. Res. Part Logist. Transp. Rev.* **2009**, *45*, 367–379.

34.   Roodbergen, K.J.; Vis, I.F.A. A model for warehouse layout. *IIE Trans.* **2006**, *38*, 799–811.

35. Roodbergen, K.J.; Sharp, G.P.; Vis, I.F.A. Designing the layout structure of manual order picking areas in warehouses. *IIE Trans.* **2008**, *40*, 1032–1045.
36. Pohl, L.M.; Meller R.D.; Gue, K.R. Turnover-based storage in non-traditional unit-load warehouse designs. *IIE Trans.* **2011**, *43*, 703–720.
37. Çelk, M.; Süral, H. Order picking under random and turnover-based storage policies in fishbone aisle warehouses. *IIE Trans.* **2014**, *46*, 283–300.
38. Li, Z.P.; Zhang, J.L.; Zhang, H.J.; Hua, G.W. Optimal selection of movable shelves under cargo-to-person picking mode. *Int. J. Simul. Model* **2017**, *16*, 145–156.
39. Jin, G.; Yang, P.; Duan, G. Multiple Deep Layout of Robotic Mobile Fulfillment System. In Proceedings of the 2020 IEEE 7th International Conference on Industrial Engineering and Applications (ICIEA), Bangkok, Thailand, 16–21 April 2020; pp. 230–234.
40. Gu, J.; Goetschalckx, M.; Mcginnis, L.F. Research on warehouse operation: A comprehensive review. *Eur. J. Oper. Res.* **2007**, *177*, 1–21.
41. Gu, J.; Goetschalckx, M.; Mcginnis, L.F. Research on warehouse design and performance evaluation: A comprehensive review. *Eur. J. Oper. Res.* **2010**, *203*, 539–549.
42. Enright, J.J.; Wurman, P.R. Optimization and coordinated autonomy in mobile fulfillment systems. Available online: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.307.716&rep=rep1&type=pdf (accessed on 28 June).
43. Kim, H.J.; Pais, C.; Shen, Z.J.M. Item Assignment Problem in a Robotic Mobile Fulfillment System. *IEEE Trans. Autom. Sci. Eng.* **2020**, *17*, 1854–1867.
44. Xie, L.; Thieme, N.; Krenzler, R.; Li, H. Introducing split orders and optimizing operational policies in robotic mobile fulfillment systems. *Eur. J. Oper. Res.* **2021**, *288*, 80–97.
45. Lamballais Tessensohn, T.; Roy, D.; De Koster, R.B.M. Inventory allocation in robotic mobile fulfillment systems. *IISE Trans.* **2020**, *52*, 1–17.
46. Wurman, P.R.; Andrea, R.D.; Mountz, M. Coordinating hundreds of cooperative, autonomous vehicles in warehouses. *Natl. Conf. Innov. Appl. Artif. Intell.* **2008**, doi:10.1609/aimag.v29i1.2082.
47. Herrero-Pérez, D and Martínez-Barberá, H. Decentralized traffic control for non-holonomic flexible automated guided vehicles in industrial environments. *Adv. Robot.* **2011**, *25*, 739–763.
48. Qi, M., Li, X., Yan, X., Zhang, C. On the evaluation of AGVS-based warehouse operation performance. *Simul. Model. Pract. Theory* **2018**, *87*, 379–394.
49. Boysen, N.; Fedtke, S.; Weidinger, F. Optimizing automated sorting in warehouses: The minimum order spread sequencing problem. *Eur. J. Oper. Res.*, **2018**, *270*, 386–400.
50. Meller, R.; Gue, K. The Application of New Aisle Designs for Unit-Load Warehouses. Available online: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.475.8030&rep=rep1&type=pdf (accessed on 28 June).