CIRRELT-2022-12

Robotic Mobile Fulfillment System with Pod Repositioning for Energy Saving

Allyson Silva
Leandro C. Coelho
Maryam Darvish
Walter Rei

April 2022

Document de travail également publié par la Faculté des sciences de l'administration de l'Université Laval, sous le numéro FSA-2022-004

Bureau de Montréal
Université de Montréal
C.P. 6 128, succ. Centre-Ville
Montréal (Québec) H3C 3J7
Tél : 1-514-343-7575
Télécopie : 1-514-343-7121

Bureau de Québec
Université Laval,
2325, rue de la Terrasse
Pavillon Palasis-Prince, local 2415
Québec (Québec) G1V 0A6
Tél : 1-418-656-2073
Télécopie : 1-418-656-2624

Robotic Mobile Fulfillment System with Pod Repositioning for Energy Saving

Allyson Silva[1,2,*], Leandro C. Coelho[1,2], Maryam Darvish[1,2], Walter Rei[1,3]

1. Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)
2. Operations and Decision Systems Department, Université Laval, Québec, Canada
3. Department of Management and Technology, Université du Québec à Montréal, Québec, Canada

Abstract. The robotic mobile fulfillment system (RMFS) allows easy inventory repositioning by returning pods to different locations after their use. A better inventory arrangement leads to an energy-efficient picking process since pods can be parked closer to where they will be requested next. After an order arrives, we have to decide which picking station will deal with it (order assignment), which pod containing the products requested should be brought to this station (pod selection), and to which storage location these pods should be returned after picking (pod repositioning). In this paper, we solve these problems in an integrated manner using a wave picking strategy, where decisions are made periodically. They are first modeled for when future demands are known. Since full information about future demands is seldom available in practice, we propose different models to solve it, such as when no information about the future demands is available (myopic approaches) or when demands are uncertain (stochastic approach). A local search matheuristic is presented to solve real-size instances. We solve the order assignment and pod selection problems and measure the energy consumption reduction when pod repositioning is integrated with them. We use a sampling scheme to represent future demands and perform detailed computational experiments for instances with real characteristics found in an RMFS. Our results attest to the value and effectiveness of considering stochastic demands when solving these operational problems.

Keywords: Robotic warehouse, mobile racking, storage location, stochastic programming, matheuristic.

Acknowledgements. The authors thank Compute Canada for providing high-performance parallel computing facilities. This project was supported by the Natural Sciences and Engineering Council of Canada (NSERC) via the discovery grants program. This support is gratefully acknowledged.

---

* Corresponding author: allyson.fernandes-da-costa-silva.1@ulaval.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2022

1. Introduction

Although many distribution center operations are still very labor-intensive, a growing effort to automate some of these processes has been observed in the last decades with the advent of e-commerce [2]. Automated warehousing systems work with manual picking stations as a product-to-picker system. A growing category of automated systems is the robotic mobile fulfillment system (RMFS), where mobile robots can lift movable inventory pods and bring them directly to stationary human pickers in fixed stations located around the storage area [5]. This system was popularized less than 15 years ago [11, 22, 28]. Since then, many providers have entered the mobile robots market and many large online retailers have switched the manual picking operations in their distribution centers to an RMFS [5, 17, 27, 33].

In addition to the intrinsic advantages associated with automation, an RMFS provides increased flexibility and scalability due to the ease of adding and removing pods and robots in the system and repositioning inventory in the storage area. They also require a relatively low investment cost, even for a large fleet of robots compared to other automated systems. Picking rates can more than double in an RMFS compared to manual warehouses since they eliminate the unproductive walking time of pickers [24]. Due to the lifting capacity of robots, the RMFS is efficient when used in warehouses containing several small and lightweight items, which makes it a perfect choice for e-commerce [2].

Early studies dealing with operational problems in the RMFS literature focused on optimizing performance measures related to the maximization of order throughput [16, 32]. Although optimizing this measure is important in scenarios where a very tight delivery deadline has to be met, in reality, most applications allow picking tasks to wait if this results in a more efficient overall picking process. For example, orders with standard shipping can be delayed over orders with priority shipping. Moreover, minimizing operation costs is as important as reducing the picking time in periods of low demand [20]. These costs are usually associated with the number of pod visits to stations, which is another common metric [1, 14, 29], or the distance traveled by robots [27]. With many robots running simultaneously in a typical warehouse, the total energy consumed is significantly high. Using energy consumption as a performance measure can balance pod visits and distances traveled by robots since energy consumption is directly related to both of them.

A common strategy to improve order picking efficiency is to process orders in batches using wave picking [8], which is adopted in real-world RMFS [31]. At the beginning of a wave, the current state of the system is known (number and capacity of operating picking stations, current set of orders in the backlog, and position of pods within the storage area). Then, several operational decisions are made, such as the assignment of orders to stations (*order assignment problem*, OAP), the selection of pods to be carried to the stations in the current wave (*pod selection problem*, PSP), and the location they should return to after picking (*pod repositioning problem*, PRP). Although these problems are known to be interrelated [21], only the OAP and the PSP have been extensively investigated in an integrated manner [14, 25, 29, 30, 31]. In wave picking, the integration of PRP decisions with the OAP and the PSP can lead to a better arrangement of inventory in the storage area for future waves. However, no study could be found that integrates decisions for all these problems, which is one of our main objectives.

We investigate different approaches to solve the integrated OAP–PSP–PRP problem in an RMFS using a wave picking strategy. The objective is to reduce the total energy consumed by robots carrying the pods around the storage area to meet the demands. We first consider an ideal scenario where the demands of future waves are known and model the integrated problem as a multi-period integer non-linear programming (MP-INLP) problem. We acknowledge that it is unrealistic to dispose of full information about future demands in most situations. Therefore, we derive new approaches to solve the integrated problem by planning waves individually. Given the information available at the beginning of each wave, in the *two-phase myopic* approach, the problem is solved using a two-phase procedure. In the first phase, the OAP and the PSP decisions are made by solving an integer linear programming (ILP) model adapted from the literature [25, 30]. Then, in the second phase, the PRP is solved using the "nearest rule", which assigns pods to the nearest available locations [21]. Still considering that no information about future demands is available, we propose the *integrated myopic* approach to solve the problem in a single phase as an integer non-linear programming (INLP) problem. Alternatively, the problem is solved considering that future demands can be predicted with uncertainty. The *stochastic* approach considers that scenarios for future demands can be sampled from an ABC distribution function commonly used in the warehousing literature, as it accounts for the skewness of demands [7]. This approach is modeled using a two-stage stochastic programming (2S-SP) model and is solved with a Benders decomposition scheme. The first stage formulates the integrated problem for the current wave, and the second stage formulates the upcoming wave represented by the sampled scenarios. We observe that the models for the integrated myopic and stochastic approaches are too heavy to solve real-case instances. Thus, we present a local search matheuristic where the two-phase myopic approach solution is improved by a simple local search to approximate optimal solutions for the other approaches.

Extensive computational experiments on instances based on the data commonly used in the RMFS literature are performed to assess the impact on energy consumption when planning a wave using all three approaches, i.e., two-phase myopic, integrated myopic, and stochastic. Their solutions are also compared against the MP-INLP solutions to observe how far they are from the best possible case. We further extend our analysis to evaluate the energy consumption when the number of pod visits is minimized and when we can delay picks until more orders arrive before planning the waves.

This paper is structured as follows. Section 2 provides a literature review on the relevant papers that solved the OAP, PSP, and PRP. Section 3 describes the RMFS considered, the decision problems, and how the energy consumption of the robots is estimated. Section 4 introduces the mathematical models for the approaches considered. Section 5 presents the local search matheuristic used to solve large instances of the integrated problem. Section 6 reports the computational results for the different approaches and situations analyzed. Finally, Section 7 presents the concluding remarks of this paper.

2. Literature Review

Nearly all studies dealing with an RMFS at the operational level, to some extent, use a solution strategy to the three problems considered (OAP, PSP, and PRP). We summarize the relevant studies that investigate these problems, focusing on those studying the PRP or at least two of the three problems considered. They are referenced in Table 1, showing the problems investigated, the picking strategy used (real-time, wave, or single solution), and the performance measures considered. More details about the RMFS are found in the recent reviews of Azadeh et al. [2], Boysen et al. [5], and Jaghbeer et al. [13].

Table 1: Studies that investigate the OAP, PSP, or PRP

Reference	*Year*	*OAP*	*PSP*	*PRP*	*Picking strategy*	*Performance measure*
Weidinger et al. [27]	2018			✓	Single solution	Distance traveled
Xiang et al. [29]	2018	✓	✓		Single solution	Pod visits
Merschformann et al. [21]	2019	✓	✓	✓	Real-time	Order throughput rate
Li et al. [19]	2020			✓	Real-time	Energy consumption
Jiang et al. [14]	2020	✓	✓		Wave	Pod visits
Rimélé et al. [23]	2021	✓	✓	✓	Real-time	Cycle and travel times
Xie et al. [30]	2021	✓	✓		Real-time, Wave	Pod visits
Valle and Beasley [25]	2021	✓	✓		Single solution	Pod visits
Aldarondo and Bozer [1]	2022		✓	✓	—	Pod visits
Zhuang et al. [31]	2022	✓	✓		Wave	Pods movements
Our study	2022	✓	✓	✓	Wave	Energy consumption

The OAP and the PSP are the problems mostly investigated simultaneously on order picking in the RMFS literature. Xiang et al. [29] solve the integrated OAP–PSP after replenishment decisions are made using an ILP model. A heuristic procedure is suggested to solve the integrated problem considering product correlations and order associations. A variable neighborhood search (VNS) is used to search for improved solutions by exchanging orders between batches. Later, Jiang et al. [14] integrate replenishment decisions made in waves with the OAP and the PSP. A divide-and-conquer paradigm is used to generate an initial solution for the problem. Then, another VNS is used to improve it. Xie et al. [30] also integrate the OAP and the PSP by proposing several ILPs where orders are allowed or forbidden to be split among different batches or periods. Although they consider real-time picking using a simulation framework, the problem is solved periodically, when some jobs are finished at the stations, such that orders are assigned in batches to the stations. A heuristic is proposed to accelerate the computational time. The PRP is solved in the simulations using a policy that positions pods in the nearest available location. They show that their integrated approach significantly reduces the number of pod visits to stations compared to when OAP and PSP decisions are made sequentially, such as in Merschformann et al. [21]. Valle and Beasley [25] also integrate OAP and PSP decisions using an ILP. Many additional constraints are proposed, such as adding picks from different pod sides, allocating a single pod to multiple stations, and balancing the workload among pickers. Two heuristics are proposed to solve the integrated problem. The first is based on the assignment of batches to one station at a time. The second fixes parts of the decision variables and solves the resulting sub-problem (partial integer optimization). After this problem is solved, they also solve the pod sequencing problem. Finally, Zhuang et al. [31] consider the OAP and the PSP and integrate them with the order and pod sequencing problems. As with most of the other mentioned studies, the objective is to minimize the number of pods visiting stations, although they consider a slightly different measure to include the number of movements between stations. The integrated problem is modeled using an ILP and, for larger instances, an adaptive large neighborhood search (ALNS) is proposed.

The PRP is considerably less investigated. Weidinger et al. [27] present an ILP model and an ALNS to solve the PRP given that the sequence of pods to bring to the stations is known. Aldarondo and Bozer [1] provide analytical formulas to determine the expected distance traveled by robots to perform a task as a function of the PSP and PRP policies, the shape of the storage area, and the locations of the stations. Li et al. [19] consider that pods are assigned to locations using a decentralized storage policy based on a turnover rate. Simulations of an RMFS operating with this policy show a significant reduction in energy consumption and an increase in the order picking efficiency. We found two studies that analyzed the three problems considered here. Merschformann et al. [21] summarize the most common policies of the literature and practice, and sequentially solve all three problems in real-time. The OAP is solved when an order is fulfilled, which triggers another order from the backlog to be assigned to the station. The PSP is solved when a robot working for a station can perform a new task. Finally, the PRP is solved when a pod leaves a station. For each of the problems, several solution policies are suggested. The RMFS operating with different combinations of policies is evaluated using a simulation framework to determine which one performs best considering several performance measures. The experiments show that cross-dependencies exist between the policies used. Rimélé et al. [23] present a mathematical framework to model the decisions considering the stochastic nature of processing times and demands. The decision process is formalized for a real-time picking strategy using a stochastic model. The model is illustrated using simple rules, similar to those used in Merschformann et al. [21].

Compared to the literature analyzed, our major contributions are summarized as: (i) we extend the ILP model from Xie et al. [30] to integrate the PRP decisions with the OAP and the PSP; (ii) we analyze the impact of integrating these problems on the energy consumption in an RMFS, since only the PRP is considered in Li et al. [19]; (iii) pod repositioning in wave picking is also new since previous studies, such as Merschformann et al. [21] and Rimélé et al. [23], considered a real-time picking strategy only; and (iv) we present the first stochastic programming model using a sampling scheme to account for the uncertainty of future demands in wave picking. The previous solution approaches either considered deterministic versions or real-time simulations to solve these problems.

3. Problem Description

In the RMFS considered, products are stored in identical storage pods. Robots can move underneath the pods, lift them, and carry them to where they are required. The storage area has a grid format, where each square represents either an aisle, used by robots to carry a pod through, or a storage location, either containing a parked pod or not. As commonly considered in the RMFS literature, both vertical and horizontal aisles are one-way and directions alternate among parallel aisles [16, 21]. Storage locations are grouped in blocks divided by rows, each with a storage location on each side, thus allowing pods to have direct access to aisles. Picking stations are equally distributed on one side of the storage area and a buffer zone separates the stations from the storage area. The ordered products are picked by stationary pickers from the pods carried to the stations. Figure 1 shows the floor plan representation of the described storage area.

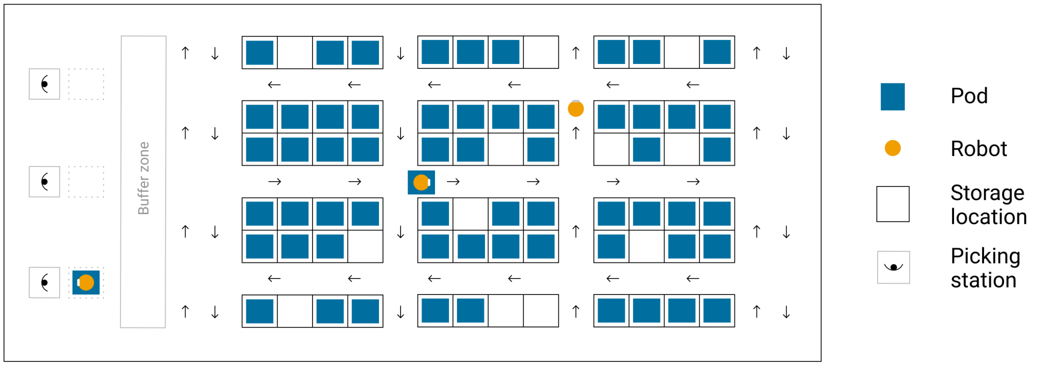


Figure 1: Representation of an RMFS storage area layout

3.1. Decision Problems

In wave picking, a planning horizon is divided into multiple periods, each representing a wave. Orders that arrived in the previous period, for example, overnight, are in a backlog and can be picked in the first wave. Each order contains a set of distinct order lines, i.e., different products to be picked. The OAP has to be solved to determine which orders will be handled by which stations [21]. In the OAP, multiple orders are batched and assigned together to stations. Overall, larger batch sizes are preferred for energy-efficient picking. However, batch sizes are limited by the capacity of stations, determined by the maximum number of bins available in the station at a time that picked products are deposited. To avoid further order consolidation operations, we consider that each bin is used to deposit products from a single order, although combining order lines is also possible when orders can be split among stations [30]. Another common consideration is to balance pickers' workload, such that each picker performs a similar number of picks in each wave [25, 31]. To account for fairness among the work distributed to pickers, we consider workload balance in our models. Workload balance is modeled such that the difference between the number of order lines assigned in a wave to all pairs of pickers does not exceed a threshold.

In a two-phase myopic approach, given an OAP solution, the next decision concerns which pods containing the demanded products should be carried to each station. Typically, scattered storage is adopted in an RMFS, such that items of the same product are spread over the warehouse in multiple pods [26]. This policy increases the probability of having some nearby pods carrying a requested product, reducing the mean processing time of orders [4, 17]. Due to the scattered storage, we have to decide which pods should be carried to the stations to meet the demands of the orders assigned to them by solving the PSP. We consider that pod replenishment is done before the beginning of the planning horizon such that the inventory in each pod is known when planning the picking waves. We also assume that a sufficient quantity of items to satisfy all orders for the planning horizon is available in each pod, which is a common and reasonable assumption in practice [31]. The initial pod locations are also known, which can be randomly generated or determined by a storage policy, such as zoning [16]. When solving the OAP and the PSP in an integrated manner, the same products contained in different orders are usually batched together and assigned to a single station. Thus, we avoid the same pod being used by multiple stations within a wave. For this reason, we also assume that each pod is carried to a station only once per wave.

Still, in the two-phase myopic approach, once the OAP and the PSP are solved, one needs to plan the robots' tasks. A task indicates where each robot should go and the path to be followed. A central server is responsible for assigning all tasks to robots. The decisions about which robots will be selected consider their positions and the arrival sequence of tasks [10, 18]. When deciding the storage location for returning a pod once the picking at a station is done, we solve the PRP [30]. When a pod is returned, it can be repositioned at any available storage location, i.e., a location that has a space to park the pod. In our models to integrate the PRP with other problems, we consider that all locations left empty by the pods demanded in the current wave are available for all pods at their return. The rearrangement of pod locations is an important aspect to be considered in a dynamic context, such as wave picking. The reason is that pods containing products that will be demanded in a future wave can be positioned near picking stations, saving time and energy.

3.2. Energy Consumption

From a context of sustainable development, the objective when solving the previously mentioned problems is to minimize the energy consumed by robots. Energy saving has been the most frequently studied topic within the context of green warehousing [3]. Typical robot's tasks in an RMFS are done following these steps:

Step 1. Move the unloaded robot from its current position to a requested pod;

Step 2. Lift the pod;

Step 3. Move the loaded robot to the designated station;

Step 4. Stop while picking is performed;

Step 5. Move the loaded robot to place the pod in its new position;

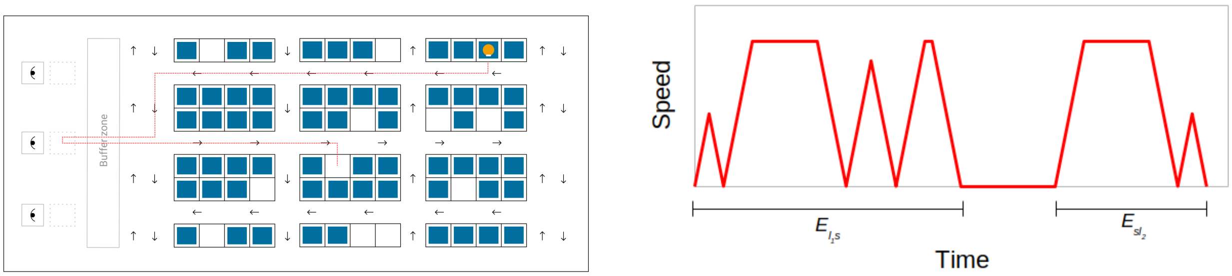
Step 6. Drop the pod.

The steps described ignore certain situations, such as blocking, obstacles that may appear in the robot's path forcing them to stop, and queues formed by robots waiting to be processed by stations. Blocking and queues are not an issue in the wave picking modeled here since time constraints are not considered, so robots carrying pods can wait until the path is entirely free at no energy cost. Unexpected obstacles rarely occur during operations such that their impact is negligible. The energy consumed to move a loaded robot is 2.5 times higher than moving an unloaded one [33]. Since most of the time robots are loaded, the effect on unloaded moves (step 1) has a low impact on the total energy consumed and is, therefore, disregarded here. For this reason, moving robots to recharging stations is also not relevant for this study. So, the total energy consumed $E_{l_1 s l_2}$ in the pertinent steps of a robot task to carry a pod from a location l_1 to a station s and back to a location l_2 is

$$E_{l_1 s l_2} = E_{lift} + E_{l_1 s} + E_{s l_2} + E_{drop}, \tag{1}$$

where the energy consumed is E_{lift} to lift the pod at its initial storage location l_1 (step 2), $E_{l_1 s}$ to move from l_1 to a station s (step 3), $E_{s l_2}$ to return the pod to a storage location l_2 (step 5), and E_{drop} to drop the pod in l_2 (step 6). The energy spent in (step 4) is negligible.

Robots paths are computed as the shortest path between storage locations and stations following the grid layout of the storage area and the directions of the aisles. To move straight between two points, a robot accelerates, reaches the maximum speed and keeps moving if the path is long enough, and then decelerates. If the path is too short, the robot does not reach its maximum speed, so the energy consumed is only a fraction of the energy spent during the acceleration/deceleration. The total energy spent in a full path between a storage location and a station, or vice-versa, is then the sum of the energy spent to perform all the straight paths contained in it. Figure 2 is based on Li et al. [19] and shows an example of the speed changes of a path traveled by a robot. Paths are short in the example when moving the pod from its storage location to the middle of the aisle where it is located, and vice-versa, and when entering in front of the station. Note in Figure 2b that speed changes are split into a section representing the path traveled by the robot when carrying the pod to the station ($E_{l_1 s}$), then it sits there while the picks are performed, and finally, it returns the pod to its new storage location ($E_{s l_2}$).



(a) Path traveled **(b)** Speed change and energy consumed

Figure 2: Path and speed of a robot to perform a task

The energy consumed by a robot traveling at a constant speed is

$$E_c = e_u \frac{d_c}{v_{max}}, \tag{2}$$

where e_u denotes the energy consumed by a unit of time, d_c is the distance traveled at a constant speed, and v_{max} is the maximum speed of a robot. The energy consumed by a robot during acceleration and deceleration is

$$E_a = E_d = e_u \frac{v_{max}}{a}, \tag{3}$$

where a is the acceleration/deceleration of a robot. In summary, $E_{l_1 s}$ and E_{sl_2} are the sum of all straight paths performed by a robot, and each straight path is the sum of E_a, E_c, and E_d, depending on its length.

Zou et al. [33] estimate the work to lift/drop a pod as $E_{lift} = E_{drop} = 0.8$ kJ. Li et al. [19] estimate that a robot charged with 2.4 kWh of energy will operate for about 6 h. Therefore, the average hourly energy consumption rate of this robot is $e_u = 0.4$ kWh or 0.4 kJ/s. They also assume a maximum speed of $v_{max} = 2$ m/s and an acceleration of $a = 1$ m/s^2. We consider that the distance between the center of each square on the square grid layout is equal to one meter. With the given equations and parameters, we can estimate the total energy consumed by a robot to perform a task.

4. Mathematical Models for the Integrated OAP–PSP–PRP

Next, we present the mathematical formulations to solve the integrated OAP–PSP–PRP problem for the RMFS previously described. For ease of reference, Table 2 provides a summary of the notation used to model the integrated problem. Other notation used throughout this section will be introduced when needed.

Table 2: Notation for the integrated OAP–PSP–PRP models

Sets	
$\mathcal{L}$	Storage locations
$\mathcal{S}$	Picking stations
$\mathcal{P}$	Pods
$\mathcal{I}$	Products
$\mathcal{P}_i \subseteq \mathcal{P}$	Pods that contain product $i \in \mathcal{I}$
$\mathcal{W} = \{1, \ldots, \lvert\mathcal{W}\rvert\}$	Waves
$\mathcal{O}$	Orders
$\mathcal{O}_w \subseteq \mathcal{O}$	Orders arrived in wave $w \in \mathcal{W}$
$\mathcal{I}_o \subseteq \mathcal{I}$	Order lines (products) of an order $o \in \mathcal{O}$
Parameters	
$E_{l_1 s l_2}$	Energy consumed by a robot to carry a pod from location l_1 to station s and return to location l_2
L_p	Initial location of pod p
C	Stations capacity (in orders) in a wave
δ	Maximum difference of order lines assigned to pickers in a wave
Decision variables	
x_{ps}^w	(PSP) Whether pod p is assigned to station s in wave w
y_{os}^w	(OAP) Whether order o is assigned to station s in wave w
y_{ios}^w	Whether product i of order o is assigned to station s in wave w
z_{pl}^w	(PRP) Whether pod p is parked at location l at the end of wave w

Regardless of the strategy used to solve the integrated problem, we are given a set of storage locations $\mathcal{L}$, picking stations $\mathcal{S}$, pods $\mathcal{P}$, and products $\mathcal{I}$. The set $\mathcal{P}_i \subseteq \mathcal{P}$ represents all pods that contain product i. The initial location of pod p is given by L_p. The maximum number of orders a station can handle in a wave is given by C. A parameter δ is used to define the maximum difference in the number of picks (order lines) performed by all pairs of pickers. Setting $\delta = 0$ imposes the same number of picks to be performed at each station. However, setting a larger δ allows more flexibility when distributing picking tasks. As discussed in Section 3.2, the energy consumed by a robot to carry a pod from a storage location l_1 to a station s and return to a location l_2 is represented by $E_{l_1 s l_2}$.

Since we consider wave picking, all decisions are made at the beginning of each wave. So, we are given $\mathcal{W} = \{1, \ldots, |\mathcal{W}|\}$ as the set of waves to be planned. Let $\mathcal{O}$ represent the set of all orders that will arrive in the planning horizon each containing the order lines (products) $\mathcal{I}_o \subseteq \mathcal{I}$. The set $\mathcal{O}_w \subseteq \mathcal{O}$ indicates all orders added to the backlog at the beginning of the wave w.

Three types of decision variables are considered, each representing one of the integrated problems. Binary variables x_{ps}^w represent the PSP decisions and indicate whether pod p is assigned to station s in wave w. Binary variables y_{os}^w represent the OAP decisions and indicate whether order o is assigned to station s in wave w. It is also required to define variables y_{ios}^{w} indicating whether product i demanded by order o is assigned to station s in wave w. Finally, binary variables z_{pl}^{w} represent the PRP decisions and indicate whether pod p is parked at location l at the end of wave w, regardless of this pod being used in this wave. For convenience, we consider z_{pl}^{0} as the binary equivalent of the parameter L_p to represent whether pod p is initially located in l.

4.1. A Multi-period Integer Non-Linear Programming Model for the Case with Known Demands

The OAP, PSP, and PRP for the RMFS presented are integrated using a MP-INLP model to consider the decisions to be made in all waves. This model works as an oracle and assumes that all orders that will arrive in each wave are known *a priori*. Although this is a strong assumption for a real-world application in e-commerce distribution centers, we will show later that this model can be adapted to be used with the data available for demands with uncertainty. The integrated problem is modeled as follows.

$$\min \sum_{w\in\mathcal{W}} \sum_{p\in\mathcal{P}} \sum_{s\in\mathcal{S}} \sum_{l_1\in\mathcal{L}} \sum_{l_2\in\mathcal{L}} E_{l_1 s l_2} z_{pl_1}^{(w-1)} x_{ps}^{w} z_{pl_2}^{w} \tag{4}$$

subject to

$$\sum_{o\in\mathcal{O}_w} \sum_{s\in\mathcal{S}} y_{os}^{w} = |\mathcal{O}_w|, \quad \forall w \in \mathcal{W} \tag{5}$$

$$\sum_{s\in\mathcal{S}} y_{os}^{w} = 1, \quad \forall w \in \mathcal{W}, o \in \mathcal{O}_w \tag{6}$$

$$y_{os}^{w} = y_{ios}^{w}, \quad \forall s \in \mathcal{S}, w \in \mathcal{W}, o \in \mathcal{O}_w, i \in \mathcal{I}_o \tag{7}$$

$$\sum_{p\in\mathcal{P}_i} x_{ps}^{w} \geq y_{ios}^{w}, \quad \forall s \in \mathcal{S}, w \in \mathcal{W}, o \in \mathcal{O}_w, i \in \mathcal{I}_o \tag{8}$$

$$\sum_{s\in\mathcal{S}} x_{ps}^{w} \leq 1, \quad \forall p \in \mathcal{P}, w \in \mathcal{W} \tag{9}$$

$$\sum_{o\in\mathcal{O}_w} y_{os}^{w} \leq C, \quad \forall s \in \mathcal{S}, w \in \mathcal{W} \tag{10}$$

$$\sum_{o\in\mathcal{O}_w} \sum_{i\in\mathcal{I}_o} \left| y_{ios_1}^{w} - y_{ios_2}^{w} \right| \leq \delta, \quad \forall s_1, s_2 \in \mathcal{S}, w \in \mathcal{W} \tag{11}$$

$$\sum_{p\in\mathcal{P}} z_{pl}^{w} \leq 1, \quad \forall l \in \mathcal{L}, w \in \mathcal{W} \tag{12}$$

$$\sum_{l\in\mathcal{L}} z_{pl}^{w} = 1, \quad \forall p \in \mathcal{P}, w \in \mathcal{W} \tag{13}$$

$$\left| z_{pl}^{(w-1)} - z_{pl}^{w} \right| \leq \sum_{s\in\mathcal{S}} x_{ps}^{w}, \quad \forall p \in \mathcal{P}, l \in \mathcal{L}, w \in \mathcal{W} \tag{14}$$

$$x_{ps}^{w} \in \{0,1\}, \quad \forall p \in \mathcal{P}, s \in \mathcal{S}, w \in \mathcal{W} \tag{15}$$

$$y_{os}^{w}, y_{ios}^{w} \in \{0,1\}, \quad \forall s \in \mathcal{S}, w \in \mathcal{W}, o \in \mathcal{O}_w, i \in \mathcal{I}_o \tag{16}$$

$$z_{pl}^{w} \in \{0,1\}, \quad \forall p \in \mathcal{P}, l \in \mathcal{L}, w \in \mathcal{W}. \tag{17}$$

The objective function (4) minimizes the energy consumed by robots to perform all tasks assigned to them during the planning horizon. The cubic expression indicates a task performed by a robot, meaning that a pod initially located in l_1 as defined in a previous wave ($z_{pl_1}^{(w-1)} = 1$) is carried in this wave to station s ($x_{ps}^{w} = 1$) and is returned to location l_2 ($z_{pl_2}^{w} = 1$). Then, the energy cost to perform this task is $E_{l_1 s l_2}$.

Constraints (5) set the number of orders satisfied in a wave equal to the number of orders arrived in that wave. Constraints (6) guarantee that orders are assigned to a single station. Constraints (7) ensure that if an order is assigned to a station in a wave, then all its order lines are also assigned to the same station in the same wave. Constraints (8) assure that at least one pod containing products assigned to a station will be carried to it when required. Constraints (9) determine that each pod can only visit one station in each wave. Capacity constraints (10) guarantee that the number of orders assigned to the stations respects their capacities. The workload balance is guaranteed by constraints (11) by imposing that the difference in the number of picks performed in each pair of stations s_1 and s_2 is lower than the threshold δ. Constraints (12)–(14) are used to solve the PRP by ensuring that pods are assigned to valid locations after each wave. Constraints (12) guarantee that no more than one pod is parked at any location after the wave. Constraints (13) guarantee that all pods are assigned to a single location at any time. Finally, constraints (14) impose that pods stay in the same location ($|z_{pl}^{(w-1)} - z_{pl}^{w}| = 0$) when not moved in a wave ($\sum_{s \in \mathcal{S}} x_{ps}^{w} = 0$). We highlight that the opposite is not always true since a pod can move and still return to the same location. The domain of the decision variables is defined in constraints (15)–(17).

4.1.1. Linearization.

The MIP model presented is non-linear due to the cubic term in the objective function and the module function in constraints (11) and (14). To make it solvable by a commercial solver for linear programming, we can linearize the former by replacing the product abc of three binary variables a, b, and c, by a new binary variable d, adding to the model the constraints $d \leq a$, $d \leq b$, $d \leq c$, and $d \geq a + b + c - 2$. Meanwhile, the latter is linearized by replacing the constraint in the form of $|a| \leq b$ by two new constraints $a \leq b$ and $a \leq -b$.

4.1.2. Valid Inequality.

A valid inequality for the MP-INLP model described consists of removing the furthest locations from the decision variables when a pod has to return from a station. The rationale is that it will never be optimal to return a pod after a pick to a storage location beyond the $|\mathcal{P}|$ closest locations from the station it was assigned to, i.e., to park in the furthest $|\mathcal{L}| - |\mathcal{P}|$ locations from this station. Since distances are station-dependent, it is not possible to simply remove a set of locations from the model. Instead, the locations are sorted by distance to each station and, then, the inequality

$$x_{ps}^{w} + z_{pl}^{w} \leq 1, \quad \forall p \in \mathcal{P}, l \in \mathcal{L}_p, s \in \mathcal{S} \tag{18}$$

is added to the model to guarantee that if pod p is assigned to station s, it cannot return to a location l in the set $\mathcal{L}_p$ containing the $|\mathcal{L}| - |\mathcal{P}|$ furthest locations from p.

4.1.3. Lower Bound.

A lower bound to the problem can be set as input to the solver to speed up the optimization process. For the problem considered, it can be computed from the minimum number of pod visits to satisfy the demands in each picking wave. To calculate a lower bound for the minimum number of pod visits in a wave, we first get the number of distinct products demanded in this wave (U). Then, we identify the pod with the maximum number of products and the number of demanded products it contains (V). Knowing U and V, a lower bound for the number of pod visits is given by $W = \lceil U/V \rceil$. An example to illustrate is as follows. If among the orders in the backlog we have to pick nine distinct products and the pod containing the most products among them has four products, we can be assured that at least $W = \lceil 9/4 \rceil = 3$ pods are required to meet all demands in this wave. Since our model minimizes energy consumption, not pod visits, we have to transform the lower bound described in terms of energy.

Given the minimum number of pod visits, a lower bound for the energy consumption is computed in a two-phase process. In the first phase, we find the W nearest pods containing at least one of the products demanded in this wave, which is a lower bound for the PSP. In the second phase, we return them to the nearest available locations, which is a lower bound for the PRP. The detailed steps for this procedure are as follows:

Step 1. Sort all pods by their distances to the nearest station;

Step 2. Get the W nearest pods in the sorted list that contain at least one product demanded in this wave;

Step 3. Add to the lower bound the distances of these pods to their nearest stations;

Step 4. Sort all available locations, including those left empty by the W nearest pods, by the nearest distance to each station;

Step 5. Given the nearest stations of the W nearest pods chosen, add to the lower bound the distances to return them to the nearest available locations.

4.1.4. Model Reduction.

Another significant improvement is possible by reducing the number of locations $|\mathcal{L}|$ in half since the energy cost to bring a pod from any row is similar, regardless of the side the pod is parked. In this case, it is enough the reformulate constraints (12) to

$$\sum_{p \in \mathcal{P}} z_{pl}^{w} \leq 2, \quad \forall l \in \mathcal{L}, w \in \mathcal{W}, \tag{19}$$

and apply it on the reduced set $\mathcal{L}$.

4.2. Solution Approaches for the Case with Demand Uncertainty

Warehouses may opt for wave picking to simplify the decision-making on the order picking process so that the operational decisions can be made periodically instead of in real-time. Wave picking also has the advantage of creating order batches more efficiently. The larger pool of orders in the backlog allows different products located in the same pod to be batched and picked together. As shown by many previous studies, this results in a reduction in pod visits but may also decrease the energy consumed by robots when pods parked in better locations are used for the picks.

In the optimization context, wave picking can be seen as a sequential decision problem in which decisions are made in an iterative process between “decide” and “reveal new information”. This strategy belongs to the *a priori optimization* modeling paradigm since the OAP, PSP, and PRP decisions are made given the current state of the warehouse considering that the uncertainty may affect the outcome. The main source of uncertainty in the RMFS lies in the future products to be picked. Therefore, in practice, warehouse managers may opt for approaches that account for the demand uncertainty to solve the integrated problem. Among the factors to be considered are the possibility of forecasting demands and the degree of difficulty to adopt the approach chosen.

The following sections describe the solution approaches considered in this study. In the *two-phase myopic* approach, we consider the most common method used in the literature where the problem is decomposed into two subproblems solved sequentially, i.e., first, the OAP and the PSP are solved together, then the PRP alone. In the *integrated myopic* approach, we assume that no information about future demands is available or that the problem is solved with no lookahead. Therefore, each wave must be planned considering only the current state of the system. Finally, in the *stochastic* approach, we assume that the information about future demands is known at a stochastic level when planning for a wave. This information can be embedded in the model, helping find robust solutions that are expected to lead to good solutions for the next wave by explicitly considering the expected future costs associated with the decisions made at the current wave.

The mathematical models for these approaches are derived from the MP-INLP model presented before. We highlight that using any of these approaches to solve the integrated problem will necessarily result in a solution worse than the optimal solution found solving the MP-INLP model. For this reason, we will refer henceforth to the integrated problem with known demands as the *integrated deterministic* (oracle) approach. Table 3 summarizes the approaches considered, showing the model implemented to represent them, the information available about future demands, and the costs considered in their objective functions.

Table 3: Summary of the solution approaches and their models for the integrated OAP–PSP–PRP

Approach	*Model*	*Future demands*	*Costs in objective function*
Integrated deterministic	MP-INLP	All waves are known	All waves
Two-phase myopic	ILP	Unknown	Bring pods in the 1^{st} wave only
Integrated myopic	INLP	Unknown	1^{st} wave only
Stochastic	2S-SP	Next wave known with uncertainty	1^{st} wave + expected future cost

4.2.1. Two-phase Myopic Approach.

Given the current wave w, in the two-phase myopic approach, the integrated problem is solved in a two-phase process using the information available for the orders currently in the backlog, i.e., the set of orders $\mathcal{O}_w$ that arrived in w. The previous MP-INLP model is decomposed into two subproblems which are solved sequentially. The first subproblem is the integrated OAP–PSP, while the second one is the PRP alone. The integrated OAP–PSP can be adapted from the MP-INLP model by removing PRP decisions. The multi-period model is reformulated to remove the index w from all decision variables. The result is an ILP formulated as:

$$\min \sum_{l\in\mathcal{L}}\sum_{s\in\mathcal{S}}(E_{lift}+E_{L_{p}s})x_{ps} \tag{20}$$

subject to

$$\sum_{o\in\mathcal{O}_w}\sum_{s\in\mathcal{S}} y_{os} = |\mathcal{O}_w|, \tag{21}$$

$$\sum_{s\in\mathcal{S}} y_{os} = 1, \quad \forall o \in \mathcal{O}_w \tag{22}$$

$$y_{os} = y_{ios}, \quad \forall s\in\mathcal{S}, o\in\mathcal{O}_w, i\in\mathcal{I}_o \tag{23}$$

$$\sum_{p\in\mathcal{P}_i} x_{ps} \geq y_{ios}, \quad \forall s\in\mathcal{S}, o\in\mathcal{O}_w, i\in\mathcal{I}_o \tag{24}$$

$$\sum_{s\in\mathcal{S}} x_{ps} \leq 1, \quad \forall p\in\mathcal{P} \tag{25}$$

$$\sum_{o\in\mathcal{O}_w} y_{os} \leq C, \quad \forall s\in\mathcal{S} \tag{26}$$

$$\sum_{o\in\mathcal{O}_w}\sum_{i\in\mathcal{I}_o} |y_{ios_1} - y_{ios_2}| \leq \delta, \quad \forall s_1, s_2 \in\mathcal{S} \tag{27}$$

$$x_{ps}, y_{os}, y_{ios} \in \{0,1\}, \quad \forall p\in\mathcal{P}, s\in\mathcal{S}, o\in\mathcal{O}_w, i\in\mathcal{I}. \tag{28}$$

The objective function (20) represents the total energy consumed by robots to bring the pods demanded from their storage locations to the stations. Constraints (21)–(27) are equivalent to constraints (5)–(11) for the OAP and the PSP decisions, and constraints (28) define the domain of the variables of the OAP and the PSP. The resulting linear model is an adaptation of the ILP model presented in Xie et al. [30] to consider energy consumption as the system performance measure and a workload balance to ensure a fair distribution of work among pickers.

This ILP model optimizes the energy consumption to bring pods to stations. However, it is still required to return pods to storage locations to complete all decisions for a wave. Since we do not consider the order of pods arriving at stations, this PRP is a simplification of the problem presented in Weidinger et al. [27]. Given the set of pods located at picking stations, defined by solving model (20)–(28) and the set of storage locations available in the storage area, a simple way to approximate the optimal solution for this PRP is by using the nearest rule presented in Merschformann et al. [21], where each pod is returned to the nearest available storage location to the station they are in.

This rule is effective since parking pods further than the nearest available location is undesirable since it will result in higher energy cost. We note that any combination of assignments of pods in a station to the nearest available storage locations results in the same PRP solution. Despite its simplicity, Merschformann et al. [21] show that the nearest rule still performed best in most cases. In our context, however, it may lead to bad solutions in the long run since it can use good locations to park pods that will not be required for a long time.

4.2.2. Integrated Myopic Approach.

The integrated myopic approach can be seen as the integrated version of the two-phase myopic approach since all three problems are solved simultaneously. Following the notation previously introduced, the integrated myopic approach is modeled as follows.

$$\min \sum_{p\in\mathcal{P}} \sum_{s\in\mathcal{S}} \sum_{l\in\mathcal{L}} E_{L_p s l} x_{ps} z_{pl} \tag{29}$$

subject to (21)–(28) and to

$$\sum_{p\in\mathcal{P}} z_{pl} \leq 1, \quad \forall l \in \mathcal{L} \tag{30}$$

$$\sum_{l\in\mathcal{L}} z_{pl} = 1, \quad \forall p \in \mathcal{P} \tag{31}$$

$$|1 - z_{pl}| \leq \sum_{s\in\mathcal{S}} x_{ps}, \quad \forall p \in \mathcal{P}, l \in \mathcal{L} \tag{32}$$

$$z_{pl} \in \{0, 1\}, \quad \forall p \in \mathcal{P}, l \in \mathcal{L}. \tag{33}$$

As in the two-phase myopic approach, the objective function and all constraints are adapted from the multi-period model to consider that decisions are made for a single wave. So, the objective function (29) considers the tasks performed to move pod p from its initial location L_p to the station it is required and back to any available location in the storage area. The integrated myopic approach considers all constraints from the two-phase myopic approach, setting the new constraints (30)–(32) for the PRP decisions. Compared to the MP-INLP model for the integrated deterministic approach, this model has two major implications. First, its size is significantly reduced without the additional index for the waves. Also, the objective function of the model is now quadratic instead of cubic, which can be linearized by replacing the quadratic term ab by a new variable c, and adding to the model the constraints $c \leq a$, $c \leq b$, and $c \geq a + b - 1$. The valid inequalities, lower bound, and model reduction described in Section 4.1 are easily adapted for the integrated myopic model considering a single wave scenario.

4.2.3. Stochastic Approach.

Given the current state of the RMFS, we have shown that the integrated myopic approach can be used to make successive decisions for the multi-period problem. However, since it only considers short-term costs, i.e., the costs incurred in the current wave, its solution can place pods with a low turnover in storage locations near the picking stations, reducing the efficiency of picking in future waves. An alternative way to solve the multi-period problem is by integrating the expected behavior of future demands arriving in the next wave to the model using stochastic programming, resulting in the so-called stochastic approach. Predicting future demands is a challenge for many warehouses using the RMFS. However, advances in tools for regression analysis, such as neural networks, are improving predictions for the short and medium terms classically made using time series estimators and other machine learning methods [9].

Consider an arbitrary solution $u^w = \{x^w, y^w, z^w\}$ for the integrated problem in the current picking wave w. The cost of u^w in the current wave for the initial pod locations z^{w-1} is represented by the function $f(u^w, z^{w-1})$. Assuming continuous distributions for the uncertain demands for the subsequent wave $w+1$, the expected cost of u^w for $w+1$ can be approximated by sampling a set Ω of scenarios to represent the backlog state at the beginning of $w+1$, where each sampled scenario $\xi \in \Omega$ has a probability of occurrence $P(\xi)$. Assuming that the scenario ξ is observed in wave $w+1$, with the pods initially arranged as defined by u^w, then the cost of the resulting solution u_ξ^{w+1} is $f(u_\xi^{w+1}, z^w)$. We ignore the expected cost of the next wave $w+2$ for the sake of avoiding the *curse of dimensionality* to allow some optimization potential. Therefore, the expected cost of wave $w+1$ for the arbitrary solutions for each backlog sampled is $\sum_{\xi \in \Omega} P(\xi) f(u_\xi^{w+1}, z^w)$. In the stochastic approach, we are interested in finding an optimal solution u^w amongst the solution space U^w representing all feasible solutions for the integrated problem in a given wave w. We do that by solving a 2S-SP model where the first stage is defined as

$$\min_{u^w \in U^w} f(u^w, z^{w-1}) + \mathbb{E}_{\xi \in \Omega}[Q(z^w, \xi)], \tag{34}$$

where $Q(z^w, \xi)$ is the optimal solution for the subproblem

$$\min_{u_\xi^{w+1} \in U_\xi^{w+1}} f(u_\xi^{w+1}, z^w), \tag{35}$$

solved in the second stage for a given scenario $\xi \in \Omega$.

The integrated myopic model presented in Section 4.2.2 is adapted to solve this 2S-SP model using a logic-based Benders decomposition technique [12]. In the first stage, it is enough to modify the objective function (29) for the integrated myopic approach to

$$\min \sum_{p\in\mathcal{P}} \sum_{s\in\mathcal{S}} \sum_{l\in\mathcal{L}} E_{L_p sl} x^w_{ps} z^w_{pl} + \sum_{\xi\in\Omega} P(\xi) Q(z^w_{pl}, \xi), \tag{36}$$

where the term $P(\xi)Q(z^w_{pl}, \xi)$ is added to represent the stochastic costs to be estimated. To this end, we solve a similar model in the second-stage to find the optimal solution for the set of orders $\mathcal{O}^\xi$ that are expected to arrive in each scenario $\xi \in \Omega$. A scenario ξ is solved for the objective function

$$Q(z^w_{pl}, \xi) = \min \sum_{p\in\mathcal{P}} \sum_{s\in\mathcal{S}} \sum_{l\in\mathcal{L}} E_{L^*_p sl} x^{w+1}_{ps} z^{w+1}_{pl}, \tag{37}$$

where L^*_p is the initial location of pod p in this stage given according to the solution found in the first stage, i.e., $L^*_p = \{l | z^w_{pl} = 1\}$. Also, the second-stage variables x^{w+1}, y^{w+1}, and z^{w+1} are subject to the integrated myopic constraints. The sampling technique used to generate $\mathcal{O}^\xi$ is presented in Section 5.2 and the samples are generated as described in Section 6.2.

We use an approach based on Benders decomposition in the solution procedure implemented to solve the 2S-SP. Assuming that we are able to enumerate the finite set of feasible solutions to the set u^w, then the term $\sum_{\xi\in\Omega} P(\xi)Q(z^w_{pl}, \xi)$ in the objective function of the first-stage model can be replaced by

$$\min c \tag{38}$$

subject to

$$c \geq \left(1 - \sum_{p\in\mathcal{P}} \sum_{l\in\mathcal{L}:\bar{z}^w_{pl}=1} (1 - z^w_{pl})\right) \sum_{\xi\in\Omega} P(\xi) Q(z^w_{pl}, \xi), \quad \forall \bar{z}^w_{pl} \text{ feasible solution for } U^w. \tag{39}$$

Constraints (39) indicate that c should be at least equal to the expected cost $\sum_{\xi\in\Omega} P(\xi)Q(z^w_{pl}, \xi)$ whenever all variables $\bar{z}^w_{pl}$, representing the z^w variables equal to 1 in the first-stage solution, reappear in the optimization process.

It is possible to improve the optimization process described using a lower bound derived for the stochastic costs. Let $LB(\xi)$ be a known lower bound for $Q(z^w_{pl}, \xi)$. The lower bound for the stochastic costs in (36) is $\sum_{\xi \in \Omega} P(\xi) LB(\xi)$. Adding this lower bound to (36) can significantly speed up the solution process since the second-stage model will be solved only when the stochastic costs can improve the current solution given the lower bound provided. With the improvement described, we must subtract $LB(\xi)$ from the objective function of the second-stage model for scenario ξ to compensate the stochastic costs already considered in the first stage. In our experiments, the lower bound $LB(\xi)$ is calculated using a modified version of the method described in Section 4.1.3 since we do not know the initial pod locations for the second stage before solving the first-stage model. Now, we consider that the minimum number of pods W to be picked will have to traverse the buffer zone twice without identifying their possible initial locations.

5. A Local Search Matheuristic for the Integrated OAP–PSP–PRP

Due to the complexity of solving most of the models presented, the design of a heuristic is required to approximate the optimal solution for the integrated problem with stochastic demands, i.e., the stochastic approach, for large-size instances. We thus design a local search matheuristic, which combines the effectiveness of generating solutions using one of the mathematical models previously introduced with a local search capable of quickly verifying a neighborhood based on the repositioning of pods.

The description of our matheuristic is divided into three parts detailed in the next sections. The first part shows how to generate a feasible solution for the problem from a simple adaptation of the two-phase myopic approach. Then, we explain how we can generate representative scenarios to estimate the stochastic cost of feasible solutions for the problem. Finally, a very efficient local search is described that can improve solutions by searching on a neighborhood defined by solutions generated by swapping pod positions in the current wave. The structure of our search allows the stochastic cost to be easily updated for each neighbor solution using simple analytical formulas.

5.1. Generating Feasible Solutions

A feasible solution for a wave requires determining which orders are assigned to which stations, which pods are assigned to which stations, and to where each pod should return after the picks, considering the constraints previously described, such as the picking stations' capacities and workload balance. The first two decisions pose the biggest challenge to generating a feasible solution since the number of existing matches between products contained in orders and pods is huge when a scattered storage is used. Previous studies attempted to design heuristics for the integrated OAP–PSP [14, 25, 29, 30].

A common technique is to use mathematical models, either entirely or partially, to obtain a feasible solution that a local search algorithm can later improve.

In this study, due to the reduced complexity of the ILP model presented for the integrated OAP–PSP in the two-phase myopic approach of Section 4.2.1 compared to the other models we have presented, we use this model to generate a set of feasible solutions for the integrated OAP–PSP. State-of-the-art mathematical programming solvers can generate a pool of solutions for a problem that are certified to be the N best solutions for it. So, we start our matheuristic by solving the ILP model until we obtain the N best solutions. Then, the PRP is solved using the nearest rule to decide where the pods should be returned to after the picks. All solutions are evaluated and only the best one is kept.

In preliminary experiments, we observed that good solutions for the integrated myopic approach are not too different than the optimal solution found in the first phase of the two-phase myopic approach. Therefore, we fixed N to 100 in all our experiments to avoid wasting time to find solutions not so close to the optimal one. Since this heuristic starts from the optimal solution of the two-phase myopic approach and possibly improves it for the current wave, it leads to an energy cost that is upper bounded by the optimal solution for the two-phase myopic approach and lower bounded by the solution for the integrated myopic approach. We show in our computational experiments (Section 6) that, despite its simplicity, our matheuristic is capable of improving the solution given by the two-phase myopic approach to near-optimal solutions for the integrated myopic approach.

5.2. Evaluating the Stochastic Cost

Given a feasible solution for the current wave, we estimate the stochastic cost of the next wave by sampling possible scenarios that may be observed in the future. We sample a number S of scenarios using a sample average approximation (SAA) scheme. The SAA is used to solve stochastic problems using a Monte Carlo simulation. It considers that a random sample of scenarios drawn from known probability distribution functions can approximate well the expected cost of all possible scenarios [15]. We use the SAA to sample a certain number of scenarios using typical demand distributions as described in Section 6.2, and the same scenarios sampled are used throughout the solution process to speed up the run.

After generating the backlog of S scenarios, we evaluate the stochastic cost by considering the pods' positions determined by the heuristic to generate feasible solutions as the initial layout. Then, we solve the ILP model for the OAP–PSP for each scenario individually followed by the PRP with the nearest rule. The stochastic cost is given by multiplying the solutions found by the probability of occurrence of each scenario.

An important remark about the evaluation of the stochastic cost using the SAA is that scenarios are evaluated independently from each other. This provides a great opportunity to perform evaluations using parallel computing. Parallelism assigns different tasks of the algorithm to different threads of a computer to speed them up, potentially linearly reducing the running time of the block of tasks being parallelized. We used parallelism when solving the OAP–PSP for each scenario, which is by far the most time-consuming task of our matheuristic.

5.3. Improving the Current Solution

Thus far, we described how a feasible solution for the current wave of the integrated OAP–PSP–PRP is generated by our matheuristic and how its stochastic cost for the subsequent wave is estimated using the SAA technique. With this information in hand, we can improve the current solution to reduce its first and second wave costs using a simple best improvement local search algorithm based on rearranging pods after they return from stations in both waves. Algorithm 1 presents the pseudocode for the local search implemented. The idea of the best improvement search is to keep updating the solution to the best one found in its neighborhood until no more improving solution can be found (lines 1 and 33). A neighborhood of a solution is defined as all the solutions that can be generated by changing the final position of a pod moved in the current wave. Given a pod p_1 moved in this wave, it can be repositioned to any available storage location, and the difference in the energy cost of this wave is $E_{s_1 l^*} - E_{s_1 l_1}$, i.e., the cost of returning this pod from the assigned station s_1 to an available location l^* instead of returning to its current location l_1 (lines 4–7). The next step is to evaluate how this movement impacts the energy cost of the future waves represented by the sampled scenarios. For each scenario, the local search updates the initial location of p_1 and its cost in case p_1 is also moved in the wave representing this scenario (lines 12–14), and searches for the pod p_2 that when repositioned to the location left empty by p_1 after the current wave leads to the lowest difference in the energy cost Δ_2^* (lines 15–21). The savings (or increase) in the total solution cost is given by the sum of the savings in both stages (line 23). If this difference is negative, then the movement of p_1 to l^* and of p_2 to l_1 results in an improving solution to the problem (lines 25–29). Since we use the best improvement strategy, we search for all solutions in the neighborhood before deciding where the solution should be moved to (line 32). The final solution for the search described is expected to approximate well the optimal solution for the stochastic approach. In the next section, we compare its results with those found by each of the previously described approaches.

Algorithm 1 Best improvement local search

repeat
  Best pod repositioning for the current wave and each scenario i: $z_1^*, z_2^* \leftarrow \emptyset$;
  Best improvement: $\Delta^* = 0$;
  for all pods p_1 moved in this wave to a station s_1 and returned to a location l_1 **do**
    for all locations available l^* after the current wave **do**
      Move p_1 from l_1 to l^* at the end of the current wave;
      $\Delta_1 = E_{s_1 l^*} - E_{s_1 l_1}$; `// Update current wave return cost`
      for all scenarios $i = \{1, \ldots, S\}$ **do**
        Δ_2^* ={Large number}
        for all pods p_2 moved in this wave to a station s_2 and returned to a location l_2 in scenario i **do**
          $\Delta_2 = 0$;
          if p_1 is also moved in i from a location l' to a station s' **then**
            $\Delta_2 = \Delta_2 + (E_{l^* s'} - E_{l' s'})$; `// Update future wave depart cost`
          end if
          if l_1 is available at the end of scenario i **then**
            Move p_2 from l_2 to l_1 at the end of the scenario i;
            $\Delta_2 = \Delta_2 + (E_{s_2 l_1} - E_{s_2 l_2})$; `// Update future wave return cost`
          end if
          if $\Delta_2 < \Delta_2^*$ **then**
            $\Delta_2^* = \Delta_2$;
          end if
        end for
        $\Delta_1 = \Delta_1 + \Delta_2^*$;
      end for
      if $\Delta_1 < \Delta^*$ **then**
        $z_1^* \leftarrow \{p_1, l^*\}$;
        $z_2^* \leftarrow \{p_2, l_1\}$;
        $\Delta^* = \Delta_1$;
      end if
    end for
  end for
  Reposition pods saved in z^* to their new locations;
until No improvement is possible ($\Delta^* \geq 0$)

6. Computational Experiments

In this section, we report and analyze the results of extensive computational experiments performed using the methods presented. Additionally, we compare solutions for a different objective function, i.e., minimizing the number of pod visits instead of the energy consumption, and we provide a new solution approach where we can wait for more orders to arrive before starting to plan the picking waves.

The computational environment used to run the experiments is equipped with an Intel Gold 6148 Skylake CPU with a 2.4 GHz clock. Runs were limited to use a maximum of 8 GB of RAM and four cores. All methods were implemented in C++, and the parallelism was implemented using OpenMP. The optimization models were solved using Gurobi 9.5.

6.1. Instance Generation

As is common in the warehousing literature, synthetic instances were generated to test the optimization models and the matheuristic presented. Their parameters are based on previous studies on the RMFS, most simulating real conditions found in warehouses. A summary of the parameters of the instances generated is presented in Table 4.

Table 4: Summary of the instances generated

Layout	$\|\mathcal{L}\|$	$\|\mathcal{S}\|$	$\|\mathcal{P}\|$	$\|\mathcal{I}\|$	$\|\mathcal{I}_p\|$	$\|\mathcal{W}\|$	$\|\mathcal{O}_w\|$	$\|\mathcal{I}_o\|$	L_p	C	δ	*Skewness*
Tiny	16	2	13	10	3	2	5	[1,4]	Rand	3	4	80%, 50%, 33%
Small	72	2	61	20, 40	5, 10	2	10	[1,4]	Rand	6	4	80%, 50%, 33%
Medium	200	3	170	50, 100	7, 15	2	25	[1,4]	Rand	10	4	80%, 50%, 33%
Large	504	4	428	200, 500	10, 25	2	50	[1,4]	Rand	15	4	80%, 50%, 33%

We generated instances in four different layout sizes – tiny, small, medium, large – each represented by different numbers of vertical aisles, horizontal aisles, and rows in each block. The tiny layout is a $2 \times 1 \times 2$, meaning that it has two vertical aisles, one horizontal aisle (front and back aisles excluded), and two rows of locations in each of its four blocks. Since each row has two storage locations (one on each side), these instances have $|\mathcal{L}| = 16$ locations where pods can be parked. Small, medium and large sizes are, respectively, $3 \times 2 \times 4$, $5 \times 4 \times 4$, and $9 \times 6 \times 6$, meaning they have 72, 200, and 504 storage locations, respectively. The largest layout generated is similar to the one used in Xie et al. [30] for their experiments to solve the OAP–PSP. The energy cost for a task $E_{l_1 s l_2}$ is set as explained in Section 3.2 for a buffer zone of five meters separating the stations from the storage area. The number of stations $|\mathcal{S}|$ is between two to four. The number of pods is equal to 85% of the number of storage locations, i.e., $|\mathcal{P}| = 0.85|\mathcal{L}|$. The number of distinct products available in the storage area $|\mathcal{I}|$ is between 10 to 500. Being $\mathcal{I}_p \subseteq \mathcal{I}$ the set of products contained in a pod p, we set $|\mathcal{I}_p|$ to be between 3 to 25 products. These combinations allow us to analyze different levels of products scatteredness in the storage area. All instances have a planning horizon of two waves. Limiting $|\mathcal{W}| = 2$ has the advantage of requiring less computational effort to solve instances and allowing a fair comparison between our solution approaches with a smaller set of instances. New waves are triggered when there are enough products in the backlog to use most of the capacity available to reduce pickers' idle time, without overloading the system and leaving some flexibility to move orders between stations when solving the OAP. The orders in each wave are generated using the procedure described in Section 6.2 for the scenarios generation such that each order has one to four products, the average order comprises 1.6 items, and the majority of them has a single item, as seen in e-commerce distribution centers. The initial location of pods L_p is determined randomly. Stations' capacities C range between 3 to 15 orders. The maximum difference of order lines picked by stations δ is arbitrarily fixed at four to allow some flexibility when assigning orders to stations. Finally, three demand skewness are considered to generate orders in all sizes, ranging from 20% of orders accounting for 80% (low), 50% (medium), and 33% (high) of all demands. The given parameters result in three settings for the tiny layout and 12 settings for the remaining layouts each. Random seeds were used to generate 20 instances for each setting, which result in 780 instances in total, all of them carefully generated to contain feasible solutions.

6.2. Scenarios Generation

A scenario contains a certain number of orders, each with a certain number of products, which can be a combination of any subset of products among all available products $\mathcal{I}$.

The first decision when sampling a scenario to estimate the stochastic costs is to determine the number of orders to be sampled. Since the starting point of a wave is a decision controlled by the warehouse manager, we simply consider that the number of orders to be generated is fixed to the number of orders arrived in the current wave.

The next decision concerns the number of order lines contained in each order. Orders in e-commerce are typically composed of few products. The number of order lines in each order is commonly generated from a truncated geometric distribution [16, 25, 30]. In this distribution, given a parameter μ the probability that an order contains m distinct products is represented by $\mu(1-\mu)^{(m-1)}/\sum_{n=1}^{M}\mu(1-\mu)^{(n-1)}$ for $m=\{1,\ldots,M\}$, where M is the maximum number of items in an order. Following Valle and Beasley [25], we generated orders using $M=4$ and $\mu=1/1.73$, chosen such that orders have an average of 1.6 order lines, which is known to be the average order demand at German Amazon warehouses [6]. The chosen distribution also implies that 85% of orders contain only one item.

There is only left to determine the lines of each order, which can be generated using an ABC curve [7]. The ABC curve is used to represent demand skewness by a continuous analytical function. The more skewed demands are, the more weight few products have in the total demand. The ABC curve is given as

$$F(x)=\frac{(1+s)x}{s+x},\quad 0\leq x\leq 1, s\geq 0, s+x\neq 0, \tag{40}$$

where x indicates the relative position of a product whose order frequency represents a fraction $F(x)$ of total warehouse activity. The parameter s indicates the skewness of the demand. In the instances generated, we used $s=0.067$ to represent the low skewness case, holding that that 20% of the products

($x = 0.2$) account for 80% of the picks ($F(x) = 0.8$), reducing to 50% when $s = 0.333$ in the medium skewness case, and to 33% when $s = 1$ in the high skewness case. Given that the available products $\mathcal{I}$ are sorted by their demands, the ABC curve is used to generate the order lines for the scenarios from random x values. Finally, equal probabilities of occurrence are assumed for each scenario among the S scenarios sampled, i.e., $P(\xi) = 1/S$.

6.3. Comparisons for Smaller Instance Sets

We start our analysis by using all methods presented, i.e., all exact methods for each approach and our matheuristic, to solve the smaller instance sets (tiny, small, and medium). Thus, we can observe what instance sizes each method is capable of solving within a reasonable time, defined here to be a maximum of one hour for each run. Four scenarios are sampled for the stochastic approach and the matheuristic to estimate the stochastic costs. In preliminary tests, sampling 1 to 32 scenarios per run, we observed that sampling four scenarios is enough to guarantee stability in the solutions provided. For this reason, we also bounded the number of cores available in each run to four to achieve high efficiency. Table 5 compares the solutions obtained when solving the sets *Tiny* to *Medium*. Column *Method* indicates whether either the exact, represented by each approach, or the matheuristic method is used. Column *Opt* shows the percentage of instances solved to optimality within the time limit established. This indicator is not shown for the matheuristic since optimality is not proven when using it. Column *Time (s)* is the average time in seconds to prove optimality using the exact models or to stop the matheuristic when it is the case. Columns *Bring pods*, 1^{st} *wave*, and *All waves* show the average energy cost to bring pods to stations in the first wave, the average total cost for the first wave, and the average total cost for both waves, respectively, for the instances solved within the time limit. We highlight in the table the best costs for each one to stress that the two-phase myopic approach optimizes the cost to bring pods in the 1^{st} wave, the integrated myopic approach optimizes all the costs for the 1^{st} wave, and the integrated deterministic approach optimizes all the costs for the whole planning horizon.

Some observations are made from Table 5. First, the MP-INLP and the 2D-SP models cannot be solved for any instance larger than those in the *Tiny* set, while the INLP model cannot be solved for any instance larger than those in the *Small* set, which significantly limits the applicability of these methods in practice. From the instances with optimality proven, we note that solving the INLP model for the integrated myopic approach is leading to solutions on average 1.1% (*Tiny*) and 3.6% (*Small*) better than solving the ILP model followed by the nearest rule for the PRP for the two-phase

Table 5: Methods comparison for the smaller instance sets

Method (approach)	*Layout*	*Opt*	*Time (s)*	*Energy*		
				Bring pods	1^{st} *wave*	*All waves*
MP-INLP (integrated deterministic)	*Tiny*	100%	85.8	9.99	20.04	**39.38**
	Small	0%	>3600	–	–	–
	Medium	0%	>3600	–	–	–
ILP + nearest rule (two-phase myopic)	*Tiny*	100%	0.1	**9.68**	19.68	39.87
	Small	100%	0.2	**14.43**	29.22	65.06
	Medium	100%	4.8	**30.47**	61.27	132.60
INLP (integrated myopic)	*Tiny*	100%	0.3	9.78	**19.45**	39.71
	Small	100%	34.0	14.99	**28.17**	61.73
	Medium	0%	>3600	–	–	–
2S-SP (stochastic)	*Tiny*	97%	155.0	9.89*	19.09*	39.09*
	Small	0%	>3600	–	–	–
	Medium	0%	>3600	–	–	–
Matheuristic (stochastic)	*Tiny*	–	0.3	9.76	19.62	39.62
	Small	–	1.5	14.80	28.53	62.19
	Medium	–	26.1	30.72	60.46	129.28

*only for the instances with optimality proven

myopic approach for the current wave being planned, which attests the effectiveness of integrating PRP decisions with the OAP and the PSP. However, solutions found by solving the INLP model for the integrated myopic approach are still 0.8% (*Tiny*) further than the best possible solutions as given by the MP-INLP model for the integrated deterministic approach. This means that there is room for improvement that could be reached using the stochastic approach. Unfortunately, the 2S-SP model for the stochastic approach could not solve all instances from any set, so a direct comparison between the costs shown in the table against any other method is not possible. Removing the instances that the 2S-SP model for the stochastic approach could not solve from the other approaches, we still observe a slight improvement of 0.04% compared to the solutions found solving the INLP for the integrated myopic approach and a deviation of 0.69% to those found solving the MP-INLP for the integrated deterministic approach.

In summary, our matheuristic is significantly improving the initial solutions generated by solving the ILP model from the two-phase myopic approach. For the *tiny* instances, we see an improvement of the average solution compared to the integrated myopic approach, which indicates that the SAA technique used is being somewhat effective. For the *small* set, although solutions are not better in the matheuristic than in the integrated myopic approach, we can see that it approximates them well in considerably less time. This time advantage becomes clear in the *medium* set when no instances are solved using the INLP for the integrated myopic approach while the matheuristic is quickly finding improvements for the initial solutions provided by solving the ILP for the two-phase myopic approach.

6.4. Comparisons for the Largest Instance Set

Now, we conduct a deeper analysis for the *large* instance set using the two methods capable of solving instances at this size, i.e., the ILP + nearest rule and the matheuristic. The results are presented in Table 6, detailed for the number of products ($|\mathcal{I}|$), the number of products per pod ($|\mathcal{I}_p|$), and the demand skewness (*skew*). Column *#Inst* shows the number of instances – out of a total of 20 instances – that the matheuristic finished its run within the time limit, allowing a direct comparison against the ILP. *Energy* is the equivalent of the column *All waves* from Table 5, and *Time (s)* reports the average run time. We also display a column *Diff. (%)* showing how much our matheuristic improves the initial solution generated by the ILP. Negative values show an improvement.

Table 6: Methods comparison for the large instance set

				ILP + nearest rule		*Matheuristic*		
$\lvert\mathcal{I}\rvert$	$\lvert\mathcal{I}_p\rvert$	*Skew*	*#Inst*	*Energy*	*Time (s)*	*Energy*	*Time (s)*	*Diff. (%)*
200	*10*	*33*	20	319.96	220.6	316.15	965.2	-1.19
		50	20	309.52	166.7	308.85	649.5	-0.22
		80	20	279.09	76.7	276.11	744.0	-1.07
	25	*33*	2	178.28	532.3	180.26	2938.4	1.11
		50	9	170.85	363.9	168.86	2593.1	-1.17
		80	14	163.36	329.2	164.24	1984.8	0.54
500	*10*	*33*	20	599.18	6.5	595.03	52.4	-0.69
		50	20	573.99	8.0	572.71	57.4	-0.22
		80	20	537.62	11.6	532.04	56.1	-1.04
	25	*33*	20	334.07	302.1	331.17	1273.7	-0.87
		50	19	322.87	243.9	320.18	1208.3	-0.83
		80	20	302.97	175.8	298.19	778.1	-1.58

From Table 6, we see that in most cases our local search matheuristic improves the solutions for the two-phase myopic approach for the real costs observed after all picking waves. This is a clear sign that solving the stochastic model – approximated here by our matheuristic – using the SAA scheme presented leads to better solutions for the integrated problem than using the two-phase myopic approach. The average improvement observed is of 0.76% for the real-size instances tested. Instances with a lower $|\mathcal{I}|$ and a higher $|\mathcal{I}_p|$ have products more scattered within the storage area and, consequently, are harder to solve due to the larger number of pod options to choose to carry to the stations. Given 428 pods in this layout size, when $|\mathcal{I}| = 200$ and $|\mathcal{I}_p| = 25$ each product can be found on average in 53.5 pods. Meanwhile, when $|\mathcal{I}| = 500$ and $|\mathcal{I}_p| = 10$ each product is stocked only in 8.6 pods on average. Despite the increased difficulty to solve, more scattered storage leads to much lower energy consumption, reducing from nearly 600 kJ per wave to around 180 kJ per wave comparing the two most extreme situations investigated and a high demand skewness. Another conclusion drawn here is that energy consumption is reduced when the demands are more skewed. This reduction is around 10.6% to 12.7% when the 20% most demanded products account for 80% of the total demand instead of when they account for only 33% of the total demand.

We highlight that not all runs of the ILS for the two-phase myopic approach finished within the time limit, indicating that this is approximately the largest instance size this method can be used in practice. Since the matheuristic starts from this solution, we cannot use it to solve larger instances either, unless a low scattered storage level is used, which is not common in an RMFS.

6.5. Further Analysis

In this section, we extend our analysis to two new cases that are worth investigating since they can significantly impact energy consumption in an RMFS. In the first case, we show how our models can be used to minimize the number of pod visits, which is a common performance measure optimized in the RMFS literature, as mentioned before, and compare solutions for this metric and the energy consumption when either of the two are minimized. The second case presents a situation where picks can be delayed so that waves are planned after the backlog has more orders than the scenarios considered so far. We compare the results for a backlog with twice the number of orders to show that, whenever possible, delaying picks can significantly save energy due to a more efficient order assignment solution.

6.5.1. Minimize the Number of Pod Visits Versus Energy Consumption.

Our exact models can be easily modified to optimize the number of pod visits instead of the energy consumed by robots. For that, it is enough to change the objective function (20) of the ILP for the integrated OAP–PSP to

$$\min \sum_{p \in \mathcal{P}} \sum_{s \in \mathcal{S}} x_{ps}. \tag{41}$$

We run the modified ILP model to analyze the trade-off between energy consumption and the number of pod visits when solving the same model for each objective, solving the PRP using the nearest rule for both cases. Table 7 summarizes the results obtained. We removed from this analysis all instances that took longer than one hour in either of the models to allow a direct comparison of the results. The number of instances compared is shown under *#Inst*. Columns *Energy* and *#Pods* show, respectively, the average energy consumption and the number of pod visits for the solutions found. Columns Δ*Energy (%)* and Δ*#Pods (%)* show the difference in each indicator when solving the problem for minimizing the number of pod visits compared to when minimizing energy consumption. Overall, the results show that minimizing the number of pod visits leads to solutions with a 9.1% to 18.6% higher energy consumption, even though between 11.7% and 25.2% fewer pods are carried to stations. Despite the savings in energy consumption, it is possible that blocking becomes more frequent, and long queues are formed at the stations when more pods are carried around, which may affect energy consumption in practice. Also, higher operational costs can occur from the larger number of robots required to pick the extra pods. These drawbacks should be weighted by the warehouse manager when deciding which metric to use.

Table 7: Comparison between minimizing energy consumption and minimizing the number of pod visits

				Min energy		*Min #Pods*			
$\lvert\mathcal{I}\rvert$	$\lvert\mathcal{I}_p\rvert$	*Skew*	*#Inst*	*Energy*	*#Pods*	*Energy*	*#Pods*	ΔEnergy (%)	Δ#Pods (%)
200	*10*	*33*	20	319.96	33.8	367.33	27.9	12.9	-21.0
		50	20	309.52	32.8	359.88	27.2	14.0	-20.6
		80	20	279.09	29.5	316.23	24.2	11.7	-21.9
	25	*33*	16	176.71	20.6	217.04	16.9	18.6	-22.2
		50	18	176.48	20.5	211.04	16.3	16.4	-25.9
		80	20	165.70	19.1	190.90	15.3	13.2	-25.2
500	*10*	*33*	20	599.18	54.1	659.75	47.4	9.2	-14.1
		50	20	573.99	51.2	631.50	45.9	9.1	-11.7
		80	20	537.62	48.9	596.46	42.9	9.9	-13.9
	25	*33*	20	334.07	33.8	379.45	28.2	12.0	-19.7
		50	20	323.70	32.7	371.81	27.7	12.9	-17.9
		80	20	302.97	31.0	341.45	26.1	11.3	-18.8

6.5.2. Picking Orders Immediately Versus Waiting Until All Orders Arrive.

Normally, the longer we wait to make decisions, the more information becomes available and the more efficient order picking can be. We present an alternative approach to solve the integrated OAP–PSP–PRP when the time available for the picks is not tight, such that we can delay the picks to be done after more orders arrive than the available capacity. We call this a *wait-and-see approach.* This approach can be seen as an integrated myopic approach with a larger number of orders in the backlog. In practice, it is preferred to wait for orders to be picked in a later period with no significant penalty in the demand satisfaction, such as when dealing with low-priority orders. In this case, more orders in the backlog allow the picks to be planned more efficiently.

We solved all instances of the large layout set, either picking orders as they arrive or using the wait-and-see approach. Despite the possibility of using the INLP model for the integrated myopic approach to solve the wait-and-see approach, we opted to use our matheuristic in this analysis so that we could compare solutions for the largest instance set, which is closer to what is found in real situations. We solve it considering that the orders for both waves are ready for picking at the beginning of the first wave. Then, we run our matheuristic limiting the number of picks to be half the number of orders in the backlog for a fair comparison against solving the original problem. After solving the problem for the first wave, we remove the orders picked and run the matheuristic again for the remaining orders. Since the wait-and-see approach generalizes the integrated myopic approach, its optimal solution is a lower bound for the sum of the optimal solutions of the integrated myopic approach for each wave. In practice, the wait-and-see approach may allow new orders to be added to the backlog as they arrive. The results found show that the wait-and-see approach leads to solutions between 16% to 17.8% cheaper than picking as orders arrive. The major drawback of this approach is the increase in order cycle times. Again, these have to be weighted when deciding which approach to use in practice.

7. Conclusions

In this paper, we investigated how the repositioning of pods in a robotic mobile fulfillment system can lead to a more efficient order picking process. This is analyzed by observing the energy consumption reduction of robots carrying pods between the storage area and the picking stations. We integrate pod repositioning decisions with those for two other operational problems commonly found in this system, namely the order assignment and the pod selection.

We proposed several approaches to solve the integrated problem using a wave picking strategy when future demands are uncertain. We showed that when pod repositioning decisions are integrated with other decisions, waves can be performed consuming up to 3.6% less energy than when decisions are made sequentially. When we add stochastic information about future demands by sampling a few scenarios and solving a two-stage stochastic programming model, solutions can be improved even further. We show that these solutions are only 0.69% on average below the best possible case when future demands are known.

We presented a local search matheuristic that starts from a solution generated by the two-phase myopic approach and improves it by searching a neighborhood with solutions where pods are returned to different locations after the pickings are done. Our matheuristic also uses information about future demands to provide robust solutions for the integrated problem. Our experiments showed that this matheuristic can find solutions up to 1.58% better than the two-phase myopic approach when instances are considerably larger than those solved by the exact methods.

Finally, further analysis showed how to adapt our methods to minimize the number of pod visits instead of energy consumption. Our experiments showed that minimizing pod visits can lead to solutions with up to 18.6% higher energy consumption compared to when energy consumption is explicitly minimized. A second case analyzed is when orders can wait to be picked in a later wave. We showed that a backlog with twice more orders can reduce energy consumption by up to 17.8%.

References

[1] F. J. Aldarondo and Y. A. Bozer. Expected distances and alternative design configurations for automated guided vehicle-based order picking systems. *International Journal of Production Research*, 60(4):1298–1315, 2022.

[2] K. Azadeh, R. B. M. De Koster, and D. Roy. Robotized and automated warehouse systems: Review and recent developments. *Transportation Science*, 53(4):917–945, 2019.

[3] M. Bortolini, E. Bottani, and H. Eric. Green warehousing: Systematic literature review and bibliometric analysis. *Journal of Cleaner Production*, 26:242–258, 2019.

[4] N. Boysen, D. Briskorn, and S. Emde. Parts-to-picker based order processing in a rack-moving mobile robots environment. *European Journal of Operational Research*, 262(2):550–562, 2017.

[5] N. Boysen, R. B. M. De Koster, and F. Weidinger. Warehousing in the e-commerce era: A survey. *European Journal of Operational Research*, 277(2):396–411, 2019.

[6] N. Boysen, K. Stephan, and F. Weidinger. Manual order consolidation with put walls: the batched order bin sequencing problem. *EURO Journal on Transportation and Logistics*, 8(2): 169–193, 2019.

[7] F. Caron, G. Marchet, and A. Perego. Routing policies and COI-based storage policies in picker-to-part systems. *International Journal of Production Research*, 36(3):713–732, 1998.

[8] E. Çeven and K. R. Gue. Optimal wave release times for order fulfillment systems with deadlines. *Transportation Science*, 51(1):52–66, 2017.

[9] R. Garnier. Concurrent neural network: a model of competition between times series. *Annals of Operations Research*, pages 1–20, 2021.

[10] A. Gharehgozli and N. Zaerpour. Robot scheduling for pod retrieval in a robotic mobile fulfillment system. *Transportation Research Part E: Logistics and Transportation Review*, 142:102087, 2020.

[11] E. Guizzo. Three engineers, hundreds of robots, one warehouse. *IEEE Spectrum*, 45(7):26–34, 2008.

[12] J. N. Hooker and G. Ottosson. Logic-based Benders decomposition. *Mathematical Programming*, 96(1):33–60, 2003.

[13] Y. Jaghbeer, R. Hanson, and M. I. Johansson. Automated order picking systems and the links between design and performance: A systematic literature review. *International Journal of Production Research*, 58(15):4489–4505, 2020.

[14] M. Jiang, K. H. Leung, Z. Lyu, and G. Q. Huang. Picking-replenishment synchronization for robotic forward-reserve warehouses. *Transportation Research Part E: Logistics and Transportation Review*, 144:102138, 2020.

[15] A. J. Kleywegt, A. Shapiro, and T. Homem-de-Mello. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12(2):479–502, 2002.

[16] T. Lamballais, D. Roy, and R. B. M. De Koster. Estimating performance in a robotic mobile fulfillment system. *European Journal of Operational Research*, 256(3):976–990, 2017.

[17] T. Lamballais, D. Roy, and R. B. M. De Koster. Inventory allocation in robotic mobile fulfillment systems. *IISE Transactions*, 52(1):1–17, 2020.

[18] J.-T Li and H.-J. Liu. Design optimization of Amazon robotics. *Automation, Control and Intelligent Systems*, 4(2):48, 2016.

[19] X. Li, G. Hua, A. Huang, J.-B. Sheu, T. C. E. Cheng, and F. Huang. Storage assignment policy with awareness of energy consumption in the Kiva mobile fulfilment system. *Transportation Research Part E: Logistics and Transportation Review*, 144:102158, 2020.

[20] P. A. Makris, A. P. Makri, and C. G. Provatidis. Energy-saving methodology for material handling applications. *Applied Energy*, 83(10):1116–1124, 2006.

[21] M. Merschformann, T. Lamballais, R. B. M. De Koster, and L. Suhl. Decision rules for robotic mobile fulfillment systems. *Operations Research Perspectives*, 6:100128, 2019.

[22] M. C. Mountz, R. D'Andrea, J. A. Laplante, P. L. I. David, P. K. Mansfield, and B. W. Amsbury. Inventory system with mobile drive unit and inventory holder. *US Patent*, 7(402):018, 2008.

[23] A. Rimélé, M. Gamache, M. Gendreau, P. Grangier, and L.-M. Rousseau. Robotic mobile fulfillment systems: A mathematical modelling framework for e-commerce applications. *International Journal of Production Research*, forthcoming:1–17, 2021.

[24] K. J. Roodbergen and I. F. A. Vis. A survey of literature on automated storage and retrieval systems. *European Journal of Operational Research*, 194(2):343–362, 2009.

[25] C. A. Valle and J. E. Beasley. Order allocation, rack allocation and rack sequencing for pickers in a mobile rack environment. *Computers & Operations Research*, 125:105090, 2021.

[26] F. Weidinger and N. Boysen. Scattered storage: How to distribute stock keeping units all around a mixed-shelves warehouse. *Transportation Science*, 52(6):1412–1427, 2018.

[27] F. Weidinger, N. Boysen, and D. Briskorn. Storage assignment with rack-moving mobile robots in KIVA warehouses. *Transportation Science*, 52(6):1479–1495, 2018.

[28] P. R. Wurman, R. D'Andrea, and M. Mountz. Coordinating hundreds of cooperative, autonomous vehicles in warehouses. *AI Magazine*, 29(1):9–9, 2008.

[29] X. Xiang, C. Liu, and L. Miao. Storage assignment and order batching problem in Kiva mobile fulfilment system. *Engineering Optimization*, 50(11):1941–1962, 2018.

[30] L. Xie, N. Thieme, R. Krenzler, and H. Li. Introducing split orders and optimizing operational policies in robotic mobile fulfillment systems. *European Journal of Operational Research*, 288(1): 80–97, 2021.

[31] Y. Zhuang, Y. Zhou, Y. Yuan, X. Hu, and E. Hassini. Order picking optimization with rack-moving mobile robots and multiple workstations. *European Journal of Operational Research*, 300 (2):527–544, 2022.

[32] B. Zou, Y. Gong, X. Xu, and Z. Yuan. Assignment rules in robotic mobile fulfilment systems for online retailers. *International Journal of Production Research*, 55(20):6175–6192, 2017.

[33] B. Zou, X. Xu, Y. Gong, and R. B. M. De Koster. Evaluating battery charging and swapping strategies in a robotic mobile fulfillment system. *European Journal of Operational Research*, 267 (2):733–753, 2018.