

RESEARCH ARTICLE OPEN ACCESS

Picking Operations in Warehouses With Dynamically Arriving Orders: How Good is Reoptimization?

Catherine Lorenz¹ | Alena Otto^{1,2} | Michel Gendreau³

¹Chair of Management Science/Operations and Supply Chain Management, University of Passau, Passau, Germany | ²Professor in Advanced Analytics in Manufacturing Management, Technical University of Munich, Heilbronn, Germany | ³Department of Mathematics and Industrial Engineering and CIRRELT, Polytechnique Montréal, Montréal, Quebec, Canada

Correspondence: Alena Otto (alena.otto@tum.de)

Received: 11 December 2024 | **Accepted:** 7 April 2025

Funding: This work was supported by Bayerische Forschungsallianz and Ministère des relations internationales et de la Francophonie.

Keywords: competitive analysis | on-line algorithms | order batching and routing problem | reoptimization | warehouses

ABSTRACT

E-commerce operations are essentially online, with customer orders arriving dynamically. However, very little is known about the performance of online policies for warehousing with respect to optimality, particularly for order picking and batching operations, which constitute a substantial portion of the total operating costs in warehouses. We aim to close this gap for one of the most prominent dynamic algorithms, namely reoptimization (*Reopt*), which reoptimizes the current solution each time a new order arrives. We examine *Reopt* in the *Online Order Batching, Sequencing, and Routing Problem (OBSRP)*, in both cases when the picker uses either a manual pushcart or a robotic cart. Moreover, we examine the *non-interventionist Reopt* in the case of a manual pushcart, wherein picking instructions are provided exclusively at the depot. We establish *analytical* performance bounds employing *worst-case and probabilistic analysis*. We demonstrate that, under generic stochastic assumptions, *Reopt* is almost surely asymptotically optimal and, notably, we validate its near-optimal performance in computational experiments across a broad range of warehouse settings. These results underscore *Reopt*'s relevance as a method for online warehousing applications.

1 | Introduction

E-commerce has emerged as an important channel of retail trade, emphasizing fast deliveries and large product assortments, while pressuring for low logistics costs. This makes warehousing an operational challenge and a leading determinant of competitiveness in the e-commerce market. The central component of efficient warehousing is order picking, which accounts for up to 60%–70% of the total operating costs [1, 2]. Most warehouses are so-called *picker-to-parts* warehouses [3, 4], in which order pickers travel around the warehouse to collect inventory items (see Figure 1). In a widespread *pick-while-sort* setup, pickers try to collect several orders simultaneously to save on costs. They use

a *manual pushcart* (Figure 1a), which needs to be taken to the depot for unloading when it is full, or a *robotic cart* (Figure 1b), which can autonomously return to the depot while the picker keeps working in the field. Each cart contains several bins, and each bin can accommodate an order. In this paper, we focus on picking operations in picker-to-parts warehouses and examine both technologies, manual pushcarts and robotic carts.

E-commerce operations are *essentially online*, with customer orders arriving dynamically. However, *very little* is known about the performance of the *online policies* for handling these orders with respect to *optimality*. A prominent online policy is *reoptimization*, which myopically optimizes picking operations for the

This is an open access article under the terms of the [Creative Commons Attribution](https://creativecommons.org/licenses/by/4.0/) License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2025 The Author(s). *Networks* published by Wiley Periodicals LLC.



Source: KBS Industrieelektronik GmbH. License: CC BY-SA 3.0

(a) Pushcart



Source: SSI Schäfer

(b) Robotic cart

FIGURE 1 | An order picker with a cart in a picker-to-parts warehouse.

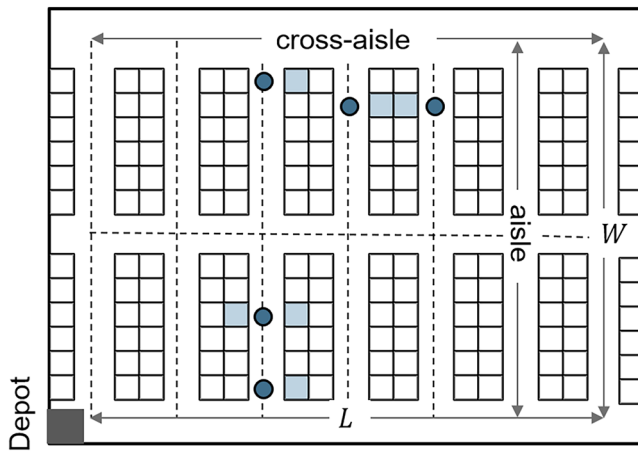


FIGURE 2 | A typical warehouse layout. A warehouse of length L and width W with three cross-aisles and seven aisles. Circles mark access points to pick highlighted items.

set of available orders at specific points in time. Moreover, reoptimization can leverage the recent progress in exact algorithms and heuristics developed for the *offline* setting, in which all the customer orders are *known* in advance (cf. [5–7]). Nevertheless, it remains open whether reoptimization makes sense for picking operations in warehouses. Optimization literature provides examples both of good [8, 9] and unfortunate (cf. [10]) performance of reoptimization in online problems.

In this paper, we study reoptimization analytically and experimentally in a generic order picking setting, which is known as the *Online Order Batching, Sequencing, and Routing Problem* (OOBSRP) in the taxonomy of Pardo et al. [11]. In OOBSRP, given the dynamically arriving orders, we have to determine (a) [batching] which orders to pick together in a single cart tour and (b) [routing] how to schedule the picking operations and how to route the picker. OOBSRP is an online routing problem set in a specialized rectilinear *network* that depicts the warehouse (see Figure 2), which includes particular capacity constraints that reflect the characteristics of order batching. In the following, Section 1.1

reviews the literature, and we state the article's contribution in Section 1.2.

1.1 | Literature Review

Online order picking and batching problems have attracted much interest in the literature, see the surveys of Boysen et al. [12], Li et al. [13], Pardo et al. [11], Van Gils et al. [14] and Vanheusden [4]. Yet, very little is known about *optimal* policies. The available research can be roughly classified into three threads. The first one analyzes intuitive policies (such as “first come first served”, S-shape routing, etc.). Here, important insights on the system stability and policy parameters are formulated using, in most cases, the tools of the queuing theory (cf. [15–19]). Within this thread Bukchin et al. [20], analyze optimal decisions on the waiting time before picking a batch for a significantly simplified problem (e.g., the time to collect a batch does not depend on the locations of the respective items in the warehouse). Secondly, a number of papers design sophisticated online policies based on metaheuristics, AI, or rule-based approaches (e.g., [21–24]). Finally, and mostly prominently, a number of authors use reoptimization as the online policy of choice (see [25–27]). In all these studies, reoptimization is performed *immediately* at the arrival of a new order. In all the discussed cases, the designed policies are usually benchmarked against simple policies, so that their *performance gap to optimality is generally unknown*.

To the best of our knowledge, only Henn [28] and Alipour et al. [29] compare selected policies to optimality. Both papers focus on the worst-case performance using *competitive analysis*. Henn [28] studies reoptimization in a special case of OOBSRP with a manual pushcart, where the picker routes start and end in the depot and follow a predefined simplified scheme, which defines the problem as the *Online Order Batching Problem* (OOBP) according to the classification of Pardo et al. [11]. In his analysis, reoptimization can only be performed when the picker retrieves a new cart in the depot; it is performed either immediately when some recently arrived orders are available or after some deliberate waiting time. The concept of waiting times seems appealing, because further orders may arrive that make better-matching batches with

the available orders. In Henn [28], the introduction of the waiting time did not improve the results of the algorithm. The paper establishes a *lower bound* of $3/2$ for the *strict competitive ratio* for any deterministic online algorithm and proves an *upper bound* of 2 for the *asymptotic competitive ratio* for their reoptimization policy (see definitions in Section 3). The tight competitive ratios remain unknown. Alipour et al. [29] extend the results of Henn [28] to multiple pickers.

On a more general note, OOBSRP can be described as a non-trivial blend of two well-known optimization problems—the *online traveling salesman problem (online TSP)* (since the picker is routed over a set of locations) and the *online batching problem (OBP)* (since the incoming orders have to be grouped into batches to be processed simultaneously in one cart tour). The seminal work of Ausiello et al. [30] introduces two versions of online TSP: *The nomadic one (N-TSP)*, for which the route of the online server ends at the last visited location, and *the homing one (H-TSP)*, which requires the server to return to the depot. As we will explain in Section 2, N-TSP and H-TSP resemble OOBSRP in the special case of infinite cart capacity in the cases where a robotic cart and a manual pushcart are used for picking, respectively. In OBP, the machine can process up to $c \in \mathbb{N}$ jobs simultaneously, and the processing time equals the processing time of the longest job. The objective is to schedule the processing of dynamically arriving non-preemptive jobs on this machine to minimize the makespan (cf. [31]). OBP can be interpreted as a special case of OOBSRP with a manual pushcart in a single-aisle warehouse with the depot located at the end of the aisle. Let us denote as *Reopt* the *immediate* reoptimization policy, which reoptimizes each time a new order arrives. The *worst-case* performance of *Reopt* has been estimated as follows: The strict competitive ratio of *Reopt* equals 2.5 for N-TSP [30], does not exceed 2.5 for H-TSP [30], and equals 2 for OBP [32]. Moreover, although better policies are possible, *Reopt*'s results in these problems are quite close to the best achievable worst-case results (see [30, 31, 33]). Further, Ascheuer et al. [33] proved that the result of *Reopt* in H-TSP is at most twice the complete-information optimum objective value plus a constant additive factor that characterizes the diameter of the studied metric space. In none of the aforementioned studies, *Reopt* was compared empirically to an optimal algorithm.

Although various extensions of the online routing problem have been studied in the literature (cf. [34–36]), only a very few papers in the literature perform probabilistic competitive analysis for online routing problems with *capacity restrictions*. The reason is that the service times of the tours formed for each capacitated vehicle by nontrivial policies, such as *Reopt*, usually have a nonzero length and are highly interdependent, which makes a formal analysis very hard. Therefore, most existing studies analyze variants of the “first come first served” policy to leverage the methods of the queuing theory (cf. e.g., [37]). An exception is a prominent article by Jaillet and Wagner [9], who studied the *online TSP with precedence and capacity constraints (OTSP-PC)*. Jaillet and Wagner [9] examine a problem extension with several salesmen, propose online algorithms with the best-possible strict competitive ratio for several problem variants, perform competitive analysis under resource augmentation, and, for specific problem variants and under several additional assumptions, perform probabilistic asymptotic analysis of online algorithms for a renewal process of arriving requests and an arrival process with

order statistics property. However, OTSP-PC studied by Jaillet and Wagner [9] differs from OOBSRP as it allows *splitting* of orders (called requests) among several tours, whereas OOBSRP does not. Consequently, once the first item of an order (called *city of a request* in OTSP-PC) is collected, a commitment to collect *all* the remaining items of this order within the current tour emerges in OOBSRP (see Example 1 of Section 3.4). This is nontrivial and changes the character of the problem completely. Overall, the idea of a probabilistic asymptotic analysis of *Reopt* as well as the nature of examined arrival processes in this paper were inspired by Jaillet and Wagner [9]. However, this paper's methodology is *distinct* in several essential ways. For instance Jaillet and Wagner [9], rely on a number of additional assumptions in their probabilistic analysis that do not apply to OOBSRP and are not used in this paper, such as Euclidean space, one item per order, and—in case of a renewal process for order arrivals—infinite growing cart capacity (called server capacity) with the number of orders n . This radically changes the mechanics of the proofs and, as in Theorem 3, requires new proof techniques.

To sum up, the optimality analysis of reoptimization for OOBSRP, which combines the features of routing (online TSP) and batching (OBP), has not been performed so far. Although the results of *Reopt* in online TSP and OBP are quite promising, it is absolutely unclear, whether it retains its good performance in the more general case of OOBSRP. Furthermore, this paper is the first to prove almost sure asymptotic optimality of a reoptimization algorithm in the case of a fixed server capacity for any capacitated online routing problem, under a renewal process model for arrival times.

1.2 | Contribution

OOBSRP, which describes picking operations in e-commerce, is a central problem in the warehousing literature. However, we still do not know the performance gap to optimality for *any* OOBSRP policy. We aim to close this gap for one of the most prominent dynamic algorithms, namely immediate reoptimization (*Reopt*), which reoptimizes the current solution each time a new order arrives. Therefore, we establish *analytical* performance bounds by means of *worst-case* and *probabilistic analysis*, and *empirically* validate the *almost optimal* performance of *Reopt*, which is indicated by our analytical findings. In particular,

- We analyze OOBSRP for two widespread technologies: Manual pushcarts and robotic picking carts. Since in the former case, transmitting instructions to the picker is only possible at the depot in some types of warehouses (cf. [38]), we also examine a so-called *non-interventionist* version of *Reopt* for manual pushcarts. *Non-interventionist Reopt* does not allow the modification of started batches, therefore, reoptimization only takes place when the picker is at the depot.
- Our main result states that *Reopt* is *almost surely asymptotically optimal* under very general stochastic assumptions in all the examined picking systems. Stated differently, in any sufficiently large instance, the result of *Reopt* coincides, with probability one, with the outcome of the optimal policy for this instance and cannot be improved. This is the case when the order arrival times can be modeled as so-called order statistics, or as a Poisson process with sufficiently small rates. In the case of a Poisson process, we also show that *Reopt*

achieves a slightly weaker notion of optimality when the order arrival rate is sufficiently large.

- In the subsequent analysis, we drop stochastic assumptions and examine the performance of *Reopt* in any large instance as well as in instances of arbitrary sizes. Among others, we prove that the worst-case performance of *Reopt* compared to the optimum (*competitive ratio*) approaches 2 in all studied picking systems. In other words, no policy can improve upon the results of *Reopt* by more than 50% in such instances.
- Our experiments, which feature typical warehouse settings and order arrival rates, reveal *almost optimal* performance of *Reopt* already in moderate-sized instances.

Our analysis offers significant insights into long-standing discussions within the warehousing literature. The first pertains to the merits of *waiting*, wherein the picker interrupts the picking process to wait in the depot (or field) for additional orders, in the hope for the arrival of better-matching orders for the next batch [11, 20]. Waiting is an integral part of prominent fixed-time-window batching and variable-time-window batching policies [11, 16]. The second involves *anticipation*, which seeks to predict upcoming order properties based on historical data [39]. In prevalent designs, anticipatory algorithms *reserve* slots in batches for future orders, resulting in the under-utilization of the cart for some time and potentially leaving out well-matching orders. Both waiting and anticipation incur immediate costs for uncertain future benefits. The performance gaps of *Reopt*, which eschews both *waiting* and *anticipation*, establish a clear benchmark for assessing the benefits of these concepts. In scenarios where *Reopt* closely approaches complete-information optimality (which is a strong concept of optimality, cf. Section 2), the introduction of waiting or anticipation cannot significantly improve the results and could potentially lead to deterioration.

We emphasize the novelty of the performed probabilistic analysis of *Reopt*. This analysis is one of the few research endeavors to scrutinize optimality gaps of a nontrivial online policy for a variant of the capacitated vehicle routing problem, diverging from the conventional “first come first served” approach, and assuming realistic capacity restrictions (see the discussion in Section 1.1).

We proceed as follows. Section 2 states OOBSRP formally, Section 3 presents a theoretical analysis of *Reopt* for OOBSRP, and Section 4 reports on computational experiments. We conclude with a discussion in Section 5.

2 | Problem Description

A typical warehouse (see Figure 2) resembles a rectilinear grid of length L and width W , which has $a \geq 1$ equidistantly positioned parallel aisles and $b \geq 2$ equidistantly positioned cross-aisles. The depot is located at some fixed location l_d in the grid. The picker moves in a rectilinear way through aisles and cross-aisles, which results in a particular distance metric $d(\cdot)$. As illustrated in Figure 2, the picker can access items only from the aisles, in particular, no items are located in cross-aisles.

Orders $(o_1, o_2, \dots, o_j, \dots)$ arrive dynamically over time and each order o_j is associated with some number $k(j) \in \mathbb{N}$ of *picking*

locations in the warehouse, where the ordered *items* have to be retrieved. Following the hierarchical planning process in warehouses, OOBSRP assumes that the picking location of each ordered item has already been determined, therefore, we use these terms interchangeably. Recall that the cart of the picker consists of $c \in \mathbb{N}$ bins, so that, to save on costs, she collects up to c orders at a time during one cart tour in a so-called *batch*. Orders cannot be split between different bins, and each bin may contain only one order at a time. Parameter c is called *batching capacity*.

We examine *picking systems* with two different types of carts: Manual pushcarts and robotic carts; we will denote these carts as *pushcart* and *robot*, respectively, in the following. The main difference between them is that the picker must bring the pushcart to the depot each time after a batch has been collected, whereas the robot can drive there autonomously, allowing the picker to proceed with picking the items “in the field” into the newly arrived robot.

OOBSRP considers one picking zone of a single picker. Therefore, when we talk about the warehouse layout below, we refer to the layout of a single zone of the warehouse.

We make the following assumptions:

- *W.l.o.g.*, we assume that the picker travels at unit speed, so that we talk about the distances and picker travel times interchangeably. In our theoretical analysis, we set the picking times of retrieving an item from the shelf and placing it in the cart to zero for simplicity. Later on, in the empirical analysis of Section 4, we consider non-zero picking times.
- Following the literature on robotic carts (e.g., [40, 41]), we assume that an empty robot is immediately available for the picker after she has completed a batch. Indeed, two robots per picker are usually sufficient to avoid picker waiting (cf. [40]).

Before stating OOBSRP in Section 2.2, we introduce the *offline* problem variant with complete information (OBSRP*) in Section 2.1. Figure 3 states an illustrative example that we use throughout this section.

2.1 | Offline Problem Variant With Complete Information: OBSRP*

An *instance* I of OBSRP* with a specific type of cart has the following input:

- Infrastructure parameters, such as the location of the depot l_d , width W , length L , the number of aisles a and cross-aisles b , as well as the batching capacity c ;
- the *sequence of dynamically arriving orders* $o = (o_1, \dots, o_n)$, where $o_j := \{s_j^1, \dots, s_j^{k(j)}\}$ refers to the set of $k(j) \geq 1$ picking locations for the j^{th} order;
- the *respective sequence of arrival times* $r = (r_1, \dots, r_n)$, with r_j being the arrival time of o_j and $r_1 \leq \dots \leq r_n$. Observe that in OBSRP*, where all the information is known in advance, arrival times can be simply interpreted as *release times* for picking the items of the corresponding orders.

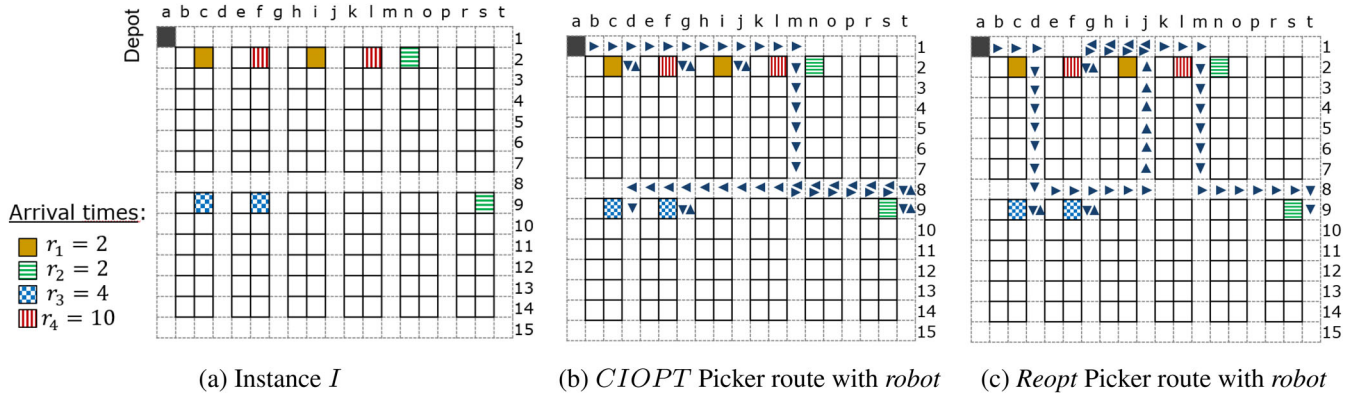


FIGURE 3 | Illustrative example. Items of all ($n = 4$) orders are marked in different colors.

TABLE 1 | Algorithms and their solutions for the illustrative instance I from Figure 3.

Algorithm	Objective value	Sequence of batches $\pi^{\text{batches}} = (S_1, S_2, \dots)$	Picker route ^b π
<i>Case of a pushcart</i>			
$CIOPT^a$	88	$(\{o_1, o_3\}, \{o_2, o_4\})$	$(a1, d2, d9, g9, j2, a1, g2, t9, m2, m2, a1)$
Non-interventionist	100	$(\{o_1, o_2\}, \{o_3, o_4\})$	$(a1, d2, j2, t9, m2, a1, d9, g9, m2, g2, a1)$
$Reopt$			
Interventionist $Reopt$	90	$(\{o_1, o_4\}, \{o_2, o_3\})$	$(a1, \mathbf{c1}, d2, \mathbf{g1}, g2, m2, j2, a1, m2, t9, g9, d9, a1)$
<i>Case of a robot</i>			
$CIOPT$	52	$(\{o_1, o_4\}, \{o_2, o_3\})$	$(a1, d2, g2, j2, m2, m2, t9, g9, d9)$
Interventionist $Reopt$	54	$(\{o_1, o_3\}, \{o_2, o_4\})$	$(a1, \mathbf{c1}, d2, \mathbf{d6}, d9, g9, j2, g2, m2, m2, t9)$

^a $CIOPT$ refers to an exact algorithm for the complete-information counterpart of instance I .

^bThe picker positions at reoptimization events of the interventionist $Reopt$ are marked in bold.

The objective is to minimize the *total completion time* of picking, which describes the picker's working time. In the system with a pushcart, the picker has to return the pushcart to the depot at the end. On the contrary, in the system with a robot, the total completion time describes the time passed until the picker picks the *last item* of the last batch. Given this objective, an optimal solution of OBSRP* for both cases of a pushcart and a robot is uniquely defined by:

- a partition of the orders into batches: $\{o_1, \dots, o_n\} = S_1 \cup S_2 \cup \dots \cup S_f, S_i \cap S_k = \emptyset \forall k, i \in \{1, \dots, f\}$; each batch contains at most c orders.
- a sequence π^{batches} of these batches, which determines the order in which the batches are processed by the picker.
- for each batch S_i , a permutation of the picking locations of all the included orders, which determines in which order the picker collects the items.

Indeed, in an optimal picking schedule, the picker will pick each item at the earliest possible time (which accounts for the arrival times of the orders) given the sequence of batches π^{batches} and the visiting sequences of picking locations for each batch. We

denote the optimal objective value of instance I as *complete information optimum* $CIOPT(I)$ and a respective optimal solution as $CIOPT$.

Note that OBSRP* is NP-hard. We refer to Lorenz et al. [42] for further details and a concise mathematical model of OBSRP* with a pushcart.

Table 1 illustrates $CIOPT(I)$ for instance I with $n = 4$ orders provided in Figure 3a in the case of a pushcart and a robot, respectively.

Note that in the optimal solutions for both cases, the picker leaves the depot towards the first picking location even before the arrival of the respective order, since the arrival times and picking locations of all the orders are known in advance. Also observe that grid cells in Figure 3 have a unit length, for example, $d(a1, d2) = 4$ as the picker has to traverse four units (cell lengths) between the cell centers of the depot $a1$ and $d2$.

In an optimal solution of OBSRP* with a pushcart, the picker arrives at each picking location after the arrival of the corresponding order. The first batch consists of two orders

($S_1 = \{o_1, o_3\}$) and is completed in $(4 + 7 + 5 + 10 + 10) = 36$ time units, after the picker's return to the depot. The second batch $S_2 = \{o_2, o_4\}$ is picked in $(7 + 19 + 13 + 0 + 13) = 52$ time units. Overall, *CIOPT* requires $(36 + 52) = 88$ time units.

Observe that in *CIOPT* for OBSRP* with a robot (cf. Figure 3b), the picker waits for the arrival of order o_4 for one time unit at location $g2$ while performing the first batch. Overall, batch $S_1 = \{o_1, o_4\}$ is picked in $(4 + 5 + [1] + 5 + 5) = 20$ time units. After having completed the first batch, the loaded robot returns to the depot autonomously, a new empty robot is immediately available, and the picker continues with the next batch $S_2 = \{o_2, o_3\}$ from her current location $m2$. *CIOPT* requires $(20 + 32) = 52$ time units.

2.2 | Online Problem and the Reoptimization Policy

In the *online* problem, OOBSP, neither the number of orders n , nor the arrival times of orders are known ahead of time, and the characteristics of each order o_j are revealed at its arrival r_j . Therefore, online algorithms represent *policies*, which prescribe a decision given the current state and available information. We analyze the online algorithm *Reopt*, which optimizes picking operations based on the available information without any anticipation of future orders, and knowledge of their probability distributions.

In traditional warehouses with a pushcart, the picker can receive instructions only at the depot, for example, because the cart and picker equipment is order-specific for each batch, so that an already commenced batch cannot be replanned. Therefore, we differentiate between *non-interventionist Reopt* and *interventionist Reopt*. In *non-interventionist Reopt*, reoptimization occurs each time the picker returns to the depot, provided newly arrived orders are available. If all the available orders have been completed, the picker simply stays at the depot until further orders arrive. Let reoptimization take place at some time t , then it amounts to solving the *complete-information* problem OBSRP* for the orders, which arrived before t and have not been picked so far; obviously, these orders are treated as immediately available.

To the contrary, in the *interventionist Reopt*, reoptimization is performed at each arrival r_j of a new order o_j . Also in this case, the planner solves a variant of OBSRP* for the already arrived, but not yet processed orders; however, she takes the following aspects into account: (i) if some bins of the cart contain picked items, the respective orders must be completed as part of the *current* batch; (ii) in the current batch, empty bins are free to be (re)assigned to any available and not yet commenced order, (iii) the picker route starts from her current position in the warehouse.

To sum up, we examine three picking systems for OOBSP, each featuring a cart technology and the applied *Reopt* algorithm: One with a pushcart and a non-interventionist *Reopt* policy (*Pcart-N*), one with a pushcart and interventionist *Reopt* (*Pcart*), and one with a robot and interventionist *Reopt* (*Robot*).

Table 1 illustrates the objective values $Reopt(I)$ of *Reopt* for the instance I from Figure 3 in the three considered picking systems: *Pcart-N*, *Pcart*, and *Robot*. In all three systems, the picker stays at the depot until the first two orders arrive at time $r_1 = r_2 = 2$.

In *Pcart-N*, at the time $t = 2$ of the first (re)optimization, only orders o_1 and o_2 are available and the batch $S_1 = \{o_1, o_2\}$ with a picking time of $(4 + 8 + 16 + 13 + 13) = 54$ is formed. Although new orders arrive at times 4 and 10, no replanning can take place until the picker returns to the depot. The next reoptimization occurs at time $(54 + 2) = 56$, and the remaining orders are assigned to the next batch. Altogether, $Reopt(I) = 56 + 44 = 100$ for *Pcart-N*.

In contrast, in *Pcart* and *Robot*, interventionist *Reopt* reoptimizes the current solution at each arrival of a new order. In both systems, at $r_3 = 4$ the picker is at decision point $c1$, and the planned routes for the batch $S_1 = \{o_1, o_2\}$ are $(a1, d2, m2, t9, j2, a1)$ and $(a1, d2, j2, m2, t9)$ in the case of *Pcart* and *Robot*, respectively. For *Pcart*, the new sequence of batches is $(\{o_1\}, \{o_2, o_3\})$ and the new route is $(a1, c1, d2, j2, a1, m2, t9, g9, d9, a1)$. This route is interrupted at the arrival of o_4 at $r_4 = 10$, when the picker is at $g1$, resulting in the final solution of Table 1 with $Reopt(I) = 90$. For *Robot*, the batching is first changed to $(\{o_1, o_3\}, \{o_2\})$ and the route is changed to $(a1, c1, d2, d9, g9, j2, m2, t9)$. The order o_4 arrives when the picker is at position $d6$, having picked the first item of o_1 . Therefore, at the next reoptimization at time $r_4 = 10$, one bin in the current batch is reserved for order o_1 . So, the new batching is $(\{o_1, o_3\}, \{o_2, o_4\})$ and results in $Reopt(I) = 33 + 21 = 54$ for *Robot* (cf. Figure 3c).

3 | Extended Competitive Analysis for Reopt in OOBSP

In this section, we perform competitive analysis for *Reopt*. In competitive analysis, the result of an algorithm $ALG(I)$ for an instance I is compared to the best possible online policy for that instance. Obviously, the best possible online policy is that of an oracle, which perfectly foresees the arrival time and the composition of each order, and the result of this policy is $CIOPT(I)$ (cf. [43]).

We apply three different ratios to highlight the various aspects of OOBSP performance. We start in Section 3.2 with the *almost surely (a.s.) asymptotical competitive ratio* $\sigma_{asympt}^{a.s.}(ALG)$, which assumes a probabilistic lens and disregards improbable events. We say that $\sigma_{asympt}^{a.s.}(ALG) = \alpha$, if:

$$\mathbb{P}(\lim_{n \rightarrow \infty} \frac{ALG(I(n)^{rand})}{CIOPT(I(n)^{rand})} = \alpha) = 1 \quad (1)$$

where $I(n)^{rand}$ is a random instance with n orders following given stochastic assumptions. Note that $\sigma_{asympt}^{a.s.}(ALG)$ strongly depends on the stochastic assumptions under consideration.

To analyze *general* large instances in Section 3.3, we turn to the *asymptotic competitive ratio* $\sigma_{asympt}(ALG)$. We call an algorithm ALG *asymptotically α -competitive* if, for any instance I , there exists a constant $const$, which is independent of the number and characteristics of the orders in I , but may depend on the warehouse geometry, such that:

$$ALG(I) \leq \alpha \cdot CIOPT(I) + const \quad (2)$$

The *asymptotic competitive ratio* $\sigma_{\text{asympt.}}(ALG)$ is then defined as the infimum over all constants α , such that ALG is asymptotically α -competitive.

Finally, to include small instances in our analysis (Section 3.4), which may display anomalies, we use the *strict competitive ratio* $\sigma(ALG)$, which is the worst-case ratio between $ALG(I)$ and $CIOPT(I)$ over all possible instances I of the problem:

$$\sigma(ALG) = \sup_I \frac{ALG(I)}{CIOPT(I)} \quad (3)$$

Observe that the strict competitive ratio derives from the asymptotic competitive ratio by forcing the constant *const* to be zero.

We begin by stating the key properties of OOBSP in Section 3.1. Afterward, Section 3.2 establishes almost sure asymptotic optimality of *Reopt* in all the examined systems under general stochastic conditions. Sections 3.3 and 3.4 examine the asymptotic and the strict competitive ratios of *Reopt*, respectively.

3.1 | Properties of OOBSP

The results of this section are used throughout the following discussion. We state bounds for a warehouse traversal in Lemma 1 as well as for the results of *Reopt* and *CIOPT* in Lemmas 2–4, respectively. We conclude by showing asymptotic optimality of *Reopt* for a special OOBSP case, where each order contains exactly *one* item.

Let us denote an OOBSP instance with n orders as $I(n)$. We denote as $I(n)^{r=0}$ the variant of some given instance $I(n)$, in which all the orders are instantly available, that is, with $r_1 = r_2 = \dots = r_n = 0$.

Lemma 1 (Upper bound for the warehouse traversal). *Notwithstanding the cart technology—a pushcart or a robot—and given an arbitrary starting position of the picker, the shortest time needed to visit all the picking locations in the warehouse and return to any given location is bounded from above by the following constant u . The bound u is tight.*

$$u = (a + 1)W + 2L \quad (4)$$

Proof. See Electronic Companion A. □

Lemma 2 (Upper bound for *Reopt*(I)). *In Pcart-N, Pcart, and Robot, the results of the respective Reopt policy for some instance $I(n)$ cannot be worse than the following bound:*

$$Reopt(I(n)) \leq r_n + CIOPT(I(n)^{r=0}) + u \quad (5)$$

Proof. At time r_n , when the last order arrives, *Reopt* reoptimizes having the complete information and requires no more time than the following simple policy: Complete the currently open batch and return to the depot in at most u time (cf. Lemma 1), then follow the route of *CIOPT* for the instance $I(n)^{r=0}$ to collect the remaining orders. □

For *Pcart* and *Pcart-N*, let us define the *makespan* $\mathcal{M}(o_j)$ of order o_j as the length of the shortest route that starts at the depot,

covers all items $s_j^i \in o_j$, and ends at the depot. Similarly, for *Robot*, we define the *makespan* $\mathcal{M}(o_j)$ of order o_j as the length of the shortest route that starts in any picking location $s_j^s \in o_j$, covers all picking locations in o_j , and ends at any arbitrary picking location $s_j^i \in o_j$.

Lemma 3. *In Pcart-N, Pcart, and Robot, the complete information optimum for any instance $I(n)^{r=0}$, for which all orders are initially available, cannot be better than the following bound:*

$$CIOPT(I(n)^{r=0}) \geq \frac{\sum_{i=1}^n \mathcal{M}(o_i)}{c} \quad (6)$$

Proof. This bound assumes the ideal case in which each batch can be fully packed with c perfectly matching orders. These orders are then collected simultaneously during the picking process at the walking cost of only one of these orders, $\mathcal{M}(o_j)$, for o_j in the batch. In the case of *Robot*, it further ignores possible walking time between the items of subsequent *distinct* batches. □

The following Lemma 4 summarizes some important straightforward relations, which we state without a proof.

Lemma 4 (Lower bounds for *CIOPT*(I)). *In Pcart-N, Pcart, and Robot, and any instance $I(n)$:*

$$CIOPT(I(n)) \geq r_n \quad (7)$$

$$CIOPT(I(n)) \geq CIOPT(I(n)^{r=0}) \quad (8)$$

$$\geq CIOPT(J^{r=0}) \quad \forall J \subseteq I(n) \quad (9)$$

Proposition 1. *In Robot, in the special case of OOBSP with single-item orders, that is, $k(o_j) = 1 \forall j \in \{1, \dots, n\}$, *Reopt* is asymptotically optimal, that is, $\sigma_{\text{asympt.}}(Reopt) = 1$.*

Proof. Observe that *Robot* with single-item orders is equivalent to the online nomadic TSP. So, at the arrival of the last order at time r_n , *Reopt* reoptimizes having the complete information and collects the remaining orders in no more time than that required by the following simple policy: Traverse the warehouse in S-shape motion once. This traversal requires a constant amount of time, dependent only on the warehouse geometry. Using (7) and the definition of the asymptotic convergence in Equation (2), we complete the proof. □

3.2 | Asymptotic Optimality of Reopt

In this section, we show that, under general stochastic assumptions, when the number of incoming orders is sufficiently large, *Reopt* is asymptotically optimal with a probability of one (*almost surely*) in all the examined systems: *Pcart-N*, *Pcart*, and *Robot*.

In a probabilistic analysis, we consider *random* instances, and such an instance with $n \in \mathbb{N}$ orders is denoted as $I(n)^{\text{rand}}$. We denote the involved random variables in capital letters, for example, orders $O = (O_1, O_2, \dots, O_n)$ and arrivals $R = (R_1, R_2, \dots, R_n)$. We assume that *distinct* customer orders O_j in O are *independently identically distributed* (i.i.d.). For mathematical completeness, we redefine O_j as a multivariate random variable

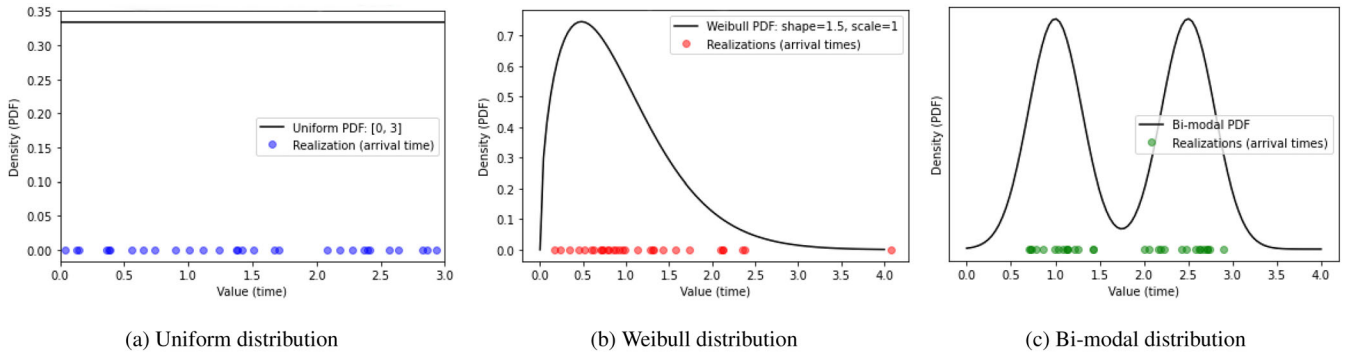


FIGURE 4 | Realizations of 30 order arrival times with the order statistics model for different distributions. The three graphs show the probability distribution function (PDF) of the respective distributions (line) and the dots on the x-axis represent the realizations of the arrival times.

$O_j = (K_j, S_j^1, \dots, S_j^{K_j})$ with K_j picking locations $S_j^1, \dots, S_j^{K_j}$. Further, in case of *Robot*, we exclude the trivial case of single-item orders (cf. Proposition 1) and assume $\mathbb{E}[K_1] > 1$. Observe that the specified stochastic assumptions explicitly allow product items S_j^i and S_j^l of the same order O_j to be correlated, such as phones and matching protective cases, which are usually ordered together.

We assume that the random sequences R and O are independent. In the following, we will discuss two common stochastic models for the arrival time sequence R .

We begin by examining the order statistics property of arrival times in Section 3.2.1, an order arrival model suggested in the context of online routing problems by Jaillet and Wagner [9]. This model considers a fixed number n of orders, with arrival times that are *independent and identically distributed (i.i.d.)* according to a fixed but *arbitrary* distribution. The *i.i.d.* assumption is natural in e-commerce warehousing, since customers act independently. The distribution can be arbitrary (see Figure 4), for example, uniform, single-peaked, skewed, multimodal (the latter applies, for instance, when most orders arrive during lunch breaks and evenings). The order statistics arrival model can have both *finite* and *infinite* support; in the former case, the orders arrive within a finite time interval, like a shift or a day, while in the latter, order arrival times are unrestricted.

For an example of the latter case, consider picking a stock of n limited-edition cell phones from a warehouse that exclusively contains those items. The picking process concludes when all n items have been picked.

The analysis in Section 3.2.2, where order arrivals follow a homogeneous Poisson process, builds upon the results of Section 3.2.1. We proceed with a formal definition of the two models and the analytical results in Sections 3.2.1 and 3.2.2.

3.2.1 | Case of the Order Statistics Property of Arrival Times

The *order statistics property of arrival times* can be defined as follows. The arrival times of the orders are i.i.d. realizations $Y_j, j \in \{1, \dots, n\}$, of a generic random variable $Y \geq 0$ with an arbitrary, given distribution and mean $\mu_Y < \infty$. To derive the arrival times of the orders, we sort the realizations of Y ; the j^{th} arrival time is the j^{th} order statistic: $R_j = Y_{(j)}$, where $Y_{(1)} \leq Y_{(2)} \leq \dots \leq Y_{(n)}$.

Theorem 1 states one of the main results of this work.

Theorem 1. *In Pcart-N, Pcart, and Robot, given the specified stochastic assumptions, including the order statistics property of arrival times:*

$$\lim_{n \rightarrow \infty} \frac{Reopt(I(n)^{rand})}{CIOPT(I(n)^{rand})} = 1 \quad \text{a.s.} \quad (10)$$

Proof. Following a similar line of proof as [9], we first recall two relations from Lemmas 4 and 2, respectively:

$$CIOPT(I(n)^{rand}) \geq CIOPT(I(n)^{rand, r=0}) \quad (11)$$

$$Reopt(I(n)^{rand}) \leq CIOPT(I(n)^{rand, r=0}) + R_n + u \quad (12)$$

Together, they imply

$$\frac{Reopt(I(n)^{rand})}{CIOPT(I(n)^{rand})} \leq 1 + \frac{n}{CIOPT(I(n)^{rand, r=0})} \cdot \frac{R_n + u}{n} \quad (13)$$

Since $\mu_Y < \infty$, Lemma 6 of [9] can be applied:

$$\lim_{n \rightarrow \infty} \frac{R_n}{n} + \frac{u}{n} = 0 \quad \text{a.s.} \quad (14)$$

At this point, the proof of [9] developed for the capacitated online TSP is not applicable, and we proceed in a unique way. To show that $\frac{n}{CIOPT(I(n)^{rand, r=0})}$ is bounded from above by a constant (which does not depend on n), we use the lower bound from Lemma 3:

$$\frac{CIOPT(I(n)^{rand, r=0})}{n} \geq \frac{1}{c} \cdot \frac{\sum_{j=1}^n \mathcal{M}(O_j)}{n} \quad (15)$$

Recall that the makespan $\mathcal{M}(O_j)$ of order O_j is the shortest picker route to collect all items in O_j . Given the stochastic assumptions, $\mathcal{M}(O_j), j \in \{1, \dots, n\}$, are i.i.d. Therefore, by the Strong Law of Large Numbers:

$$\lim_{n \rightarrow \infty} \frac{1}{c} \cdot \frac{\sum_{j=1}^n \mathcal{M}(O_j)}{n} = \frac{1}{c} \cdot \mathbb{E}[\mathcal{M}(O_1)] > 0 \quad \text{a.s.} \quad (16)$$

We only consider the case of a strictly positive $\mathbb{E}[\mathcal{M}(O_1)]$. Otherwise, the problem is trivial, and *Reopt* is optimal per definition in *Pcart-N* and *Pcart*; in *Robot*, the problem reduces to the single-item order case for which *Reopt* is asymptotically optimal by Proposition 1.

Observe that $\frac{1}{c} \cdot \mathbb{E}[\mathcal{M}(O_1)]$ is a constant that does not depend on n . The convergence of (16) implies that the right-hand side of (15) is larger than some *nonzero* constant $\zeta < \frac{1}{c} \mathbb{E}[\mathcal{M}(O_1)]$ a.s. if the number of orders n is large enough, that is, $\forall n > n_0$ for some n_0 :

$$\frac{CLOPT(I(n)^{rand, r=0})}{n} \geq \zeta \Leftrightarrow \frac{n}{CLOPT(I(n)^{rand, r=0})} \leq \frac{1}{\zeta} \quad \text{a.s.} \quad (17)$$

This, together with (13) and (14), completes the proof:

$$\begin{aligned} \frac{Reopt(I(n)^{rand})}{CLOPT(I(n)^{rand})} &\leq 1 + \frac{n}{CLOPT(I(n)^{rand, r=0})} \cdot \frac{R_n + u}{n} \\ &\xrightarrow{n \rightarrow \infty} 1 + 0 \quad \text{a.s.} \end{aligned}$$

□

The interpretation of Theorem 1 for an order statistic with a *finite*-support arrival-time distribution Y , such as during a shift, is quite intuitive. As the number of orders arriving during this shift increases, the picker becomes overloaded, and most orders will have to be picked *after* the shift ends, regardless of the picking policy. Therefore, *Reopt*, which picks optimally after the end of the shift when no more orders arrive, converges to optimality. However, the interpretation is less straightforward for a random arrival-time distribution Y with *infinite* support, such as a normal distribution in the limited-edition cell phone example in Section 3.2. In this case, as the number of available products increases, $n \rightarrow \infty$, the arrival time of the last order placement, R_n , will extend to infinity (as for arbitrary $x < \infty$, $\mathbb{P}(R_n > x) = 1 - \mathbb{P}(Y \leq x)^n \rightarrow 1$ by definition of Y 's infinite support). This implies that the picking system remains *dynamic* throughout the entire picking process. Even for these systems, Theorem 1 proves that *Reopt* can keep up with an optimal picking strategy, with probability one.

3.2.2 | Case of the Homogeneous Poisson Process for Order Arrivals

Let random variables $X_j, j \in \mathbb{N}$, which represent the time between the j^{th} and the $(j-1)^{th}$ order arrival, be i.i.d. exponentially distributed with mean $\frac{1}{\lambda}$, then the sequence of order arrival times follows a *homogeneous Poisson process* with rate λ .

A central notion for the analysis of the Poisson-arriving orders is the *queue* $Q^{Reopt}(I(n)^{rand})$, which is the set of pending orders at time R_n in instance $I(n)^{rand}$ for policy *Reopt*.

We prove in Theorem 2 that *Reopt* is a.s. asymptotically optimal if $\lim_{n \rightarrow \infty} \frac{|Q^{Reopt}(I(n)^{rand})|}{n} = 0$, that is, if with probability one, the queue of pending orders does not grow as fast as the total number of received orders. Note that this condition would be necessary under the *Reopt* policy to safeguard the warehousing system against potential instability.

Because the queue length $|Q^{Reopt}(I(n)^{rand})|$ is generated by two independent sequences of i.i.d. random variables—the picking locations and the inter-arrival times—we believe that by the 0-1-Laws, the probability of the event $\lim_{n \rightarrow \infty} \frac{|Q^{Reopt}(I(n)^{rand})|}{n} = 0$ is either 1 or zero for a given rate λ , and we suspect that the

probability increases monotonously in λ . In other words, we believe that $\lim_{n \rightarrow \infty} \frac{|Q^{Reopt}(I(n)^{rand})|}{n} = 0$ is true for small arrival rates until a fixed threshold $\bar{\lambda}$ (see Conjecture 1).

In the case of asymptotically increasing arrival rates, the probability that *Reopt* approaches optimality converges to 1 as well, as we prove in Theorem 3.

Observe the *stand-alone character* of this section's arguments in the warehousing research. The available queuing systems analysis, which is usually employed when order arrivals follow a Poisson process, requires i.i.d. service times of the batches. The latter is the case, for instance, for the “first come first served” policy, when orders are serviced in the sequence of their arrival. This is not the case for *Reopt* and neither it is the case for *CLOPT*, where the service times of the batches are interdependent due to reoptimization.

Theorem 2. In *Pcart-N*, *Pcart*, and *Robot*, if $\lim_{n \rightarrow \infty} \frac{|Q^{Reopt}(I(n)^{rand})|}{n} = 0$ a.s., then:

$$\lim_{n \rightarrow \infty} \frac{Reopt(I(n)^{rand})}{CLOPT(I(n)^{rand})} = 1 \quad \text{a.s.} \quad (18)$$

Proof. Exploiting the i.i.d. inter-arrival time property of the Poisson process, the proof combines the Strong Law of Large Numbers with the fact that the warehouse has bounded dimensions. We refer to the Electronic Companion B for details. □

Observe that Theorem 2 demonstrates the asymptotic optimality of *Reopt* not only for a.s. *stable* picking systems, but also for certain systems with high-order arrival rates, where there is a non-zero probability that the queue length grows to infinity over time. For instance, it applies to picking systems with queues growing as fast as $\sqrt{n}$, which is sublinear in n . In summary, the statement of Theorem 2 is quite general, extending from small to moderately high-order arrival rates.

Conjecture 1. In *Pcart-N*, *Pcart*, and *Robot*, there is a rate $\bar{\lambda}$ such that $\mathbb{P}(\lim_{n \rightarrow \infty} \frac{|Q^{Reopt}(I(n)^{rand})|}{n} = 0) = 1$ for all $\lambda < \bar{\lambda}$. Thus, for all rates $\lambda < \bar{\lambda}$:

$$\lim_{n \rightarrow \infty} \frac{Reopt(I(n)^{rand})}{CLOPT(I(n)^{rand})} = 1 \quad \text{a.s.} \quad (19)$$

It remains unclear whether *Reopt* is asymptotically optimal a.s. for rates $\lambda \geq \bar{\lambda}$. To prove a somewhat weaker notion of convergence, we fix the time interval $[0, t]$ (e.g., a shift) and examine the orders $I(t)^{rand}$ arrived in this time interval $[0, t]$ given the arrival rate λ of the underlying Poisson process. Recall that, by definition, the number of orders $N(t)$ in the interval is Poisson($\lambda \cdot t$)-distributed. We denote by $\mathbb{P}_\lambda$ the underlying probability measure for a given arrival rate λ .

Theorem 3. Let $\lambda_i > 0, i \in \mathbb{N}$ be any increasing sequence of arrival rates such that $\lambda_i \rightarrow \infty$ if $i \rightarrow \infty$. In *Pcart-N*, *Pcart*, and *Robot*, the following holds true:

$$\lim_{i \rightarrow \infty} \mathbb{E}_{\lambda_i} \left[\frac{Reopt(I(t)^{rand})}{CLOPT(I(t)^{rand})} \right] = 1 \quad (20)$$

And for all $\epsilon > 0$:

$$\lim_{i \rightarrow \infty} \mathbb{P}_{\lambda_i} \left(\frac{Reopt(I(t)^{rand})}{CIOPT(I(t)^{rand})} \geq 1 + \epsilon \right) = 0 \quad (21)$$

Proof. The detailed proof is provided in the Electronic Companion C.

Recall that to establish convergence, Theorem 1 depended on a sublinear growth of the arrival time R_n of the n^{th} order ($\frac{R_n}{n} \xrightarrow[n \rightarrow \infty]{a.s.} 0$). This is not the case when the arrival-time sequence is a Poisson process with some given inter-arrival time $\frac{1}{\lambda}$. Therefore, we must resort to a very different line of proof. In this proof, we perform a formal transition from the probability space described in Section 3.2.1 to the probability space of this section. The proof proceeds in three steps.

First, we want to leverage the convergence statement of Theorem 1. Therefore, we apply it in the case of instances $I(n)^{rand}$ with independent uniformly $[0, t]$ -distributed arrival times for an arbitrary $t \in \mathbb{R}^+$. Since the ratio $\frac{Reopt(I(n)^{rand})}{CIOPT(I(n)^{rand})}$ is bounded (cf. Section 3.4), we can restate Theorem 1 as the *convergence in mean*.

Secondly, we assume that the order arrival times follow a Poisson process with some given arrival rate $\lambda \in \mathbb{R}^+$. We examine instances $I(t)^{rand}$ comprising all orders arriving within a predefined time interval $[0, t]$, $t \in \mathbb{R}^+$. Let $N(t)$ be the random number of these orders. We then use the Order Statistics Property of the homogeneous Poisson process, in combination with the statements of the first step of this proof, to derive the *conditional convergence in mean*:

$$\lim_{n \rightarrow \infty} \mathbb{E}_{\lambda} \left[\frac{Reopt(I(t)^{rand})}{CIOPT(I(t)^{rand})} \mid N(t) = n \right] = 1 \quad (22)$$

Convergence implies that for any $\epsilon > 0$, we can find $n_{\epsilon} \in \mathbb{N}$ such that $\mathbb{E}_{\lambda} \left[\frac{Reopt(I(t)^{rand})}{CIOPT(I(t)^{rand})} \mid N(t) = n \right] \leq 1 + \frac{\epsilon}{2}$ for all $n > n_{\epsilon}$.

Thirdly, we compute the *unconditional* expectation $\mathbb{E}_{\lambda} \left[\frac{Reopt(I(t)^{rand})}{CIOPT(I(t)^{rand})} \right]$ using the Law of Total Expectation for distinct groups of instances—those with $N(t) > n_{\epsilon}$ and those with $N(t) \leq n_{\epsilon}$. After some transformations, we show that for any sequence of rates λ_i with $\lambda_i \xrightarrow{i \rightarrow \infty} \infty$ and any $\epsilon > 0$, one can choose $i_{\epsilon} \in \mathbb{N}$ in dependence of t and ϵ , such that $\mathbb{E}_{\lambda_i} \left[\frac{Reopt(I(t)^{rand})}{CIOPT(I(t)^{rand})} \right] \leq (1 + \epsilon)$ for all $i \geq i_{\epsilon}$.

We use Markov's inequality to derive the implication of (21) from Equation (20). $\square$

3.3 | Performance of Reopt in General Large OOBSP Instances

In this section, we abolish the stochastic assumptions and examine whether *Reopt* retains its good performance for general large problem instances. We show that even in the *worst-case*, the *Reopt* result cannot be improved by more than 50% in any given OOBSP instance if it is sufficiently large. This is still an excellent performance.

Overall, having examined different worst-case examples for *Reopt* in OOBSP with a pushcart and a robot, we see two underlying

mechanisms of the *Reopt*'s suboptimal result: (i) unfortunate batching and, in the case of the interventionist *Reopt*, (ii) unnecessary walking. In (i), *Reopt* places orders in one batch even if the savings from their batching are small compared to a separate picking of these orders (cf. Example 1). *CIOPT*, to the contrary, can anticipate upcoming better-matching orders. In (ii), *Reopt* completely changes the current route of the picker, including her next destination, at the slightest promise of time-saving. Paradoxically, this behavior may result in a prolonged back-and-forth walking without picking an item, which is avoided in *CIOPT* (cf. the proof of Proposition 3).

In the following, Proposition 2 establishes the upper bound of 2 for $\sigma_{asympt.}(Reopt)$ in all the examined picking systems. Proposition 3 states that this worst-case ratio is *tight* for *Pcart* and *Robot*.

Proposition 2. In *Pcart-N*, *Pcart*, and *Robot*, $\sigma_{asympt.}(Reopt) \leq 2$. In other words, for every instance $I(n)$

$$Reopt(I(n)) \leq 2 \cdot CIOPT(I(n)) + const \quad (23)$$

where *const* is the warehouse traversal time u from Lemma 1.

Proof. By Lemmas 2 and 4:

$$\begin{aligned} Reopt(I(n)) &\leq r_n + u + CIOPT(I(n)^{r=0}) \\ &\leq 2 \cdot CIOPT(I(n)) + u \end{aligned} \quad (24)$$

$\square$

Proposition 3. In *Pcart* and *Robot*, the upper bound of 2 is tight for the asymptotic competitive ratio, that is,

$$\sigma_{asympt.}(Reopt) \geq 2 \quad (25)$$

Note that in the case of *Pcart*, the statement requires a warehouse with at least two aisles.

Proof. We examine the case of *Pcart* and refer to Electronic Companion E for the case of *Robot*.

Figure 5 provides an unfortunate example I^{worst} for $c = 2$ and a two-aisle warehouse. This example can be generalized for $c > 2$ and more general warehouse layouts.

Arriving orders in I^{worst} consist of one item, which is placed either in the left aisle (at position s^{left}) or in the right aisle (at

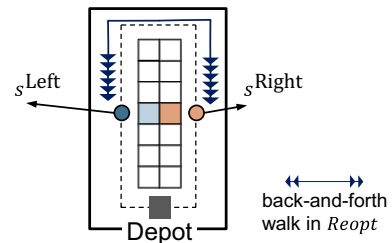


FIGURE 5 | *Pcart*: Unfortunate large instance for *Reopt*. $c = 2$. Single-item orders with items s^{left} and s^{right} arrive such that, in *Reopt*, the picker keeps walking back and forth without completing any batch until the arrival of the last order.

position s^{right}). We simplify the notation for these single-order items and write $o_i = s^{\text{left}}$ or $o_i = s^{\text{right}}, \forall i \in \mathbb{N}$, respectively. Recall that W and L are the width and the length of the warehouse, respectively. In this example, the picker can traverse both aisles in a cyclic tour of length $T = 2W + 2L$ from the depot. Let $d(l_d, s^{\text{left}}) = (l_d, s^{\text{right}}) = \frac{1}{2} \cdot d(s^{\text{left}}, s^{\text{right}}) = \frac{1}{2}T$.

Instance I^{worst} consists of $n = 2k$ orders, where $k \in \mathbb{N}_{\geq 4}$ is some *uneven* number. Each time, orders s^{left} and s^{right} arrive simultaneously. The idea is to set the inter-arrival times of the orders such that *Reopt* makes the picker oscillate between the positions of s^{left} and s^{right} in the upper half of the warehouse, without picking any item, until the last order arrives.

At time $r_1 = r_2 = 0$, *Reopt* batches the only two available orders together: $S_1 = \{o_1, o_2\}$ with $o_1 = s^{\text{left}}$ and $o_2 = s^{\text{right}}$, and plans to pick them in a circular tour. *W.l.o.g.*, let *Reopt* pick o_1 first (at time $\frac{1}{4}T$).

At time $r_3 = r_4 = \frac{3}{4}T - \frac{\delta}{2k-2}$, an order s^{left} and an order s^{right} arrive. In *Reopt*, the picker is currently at distance $\frac{\delta}{2k-2}$ above item s^{right} with o_1 placed in her cart. *Reopt* has two choices. The first is to follow the initial plan and pick batch $\{o_1, o_2\} = \{s^{\text{left}}, s^{\text{right}}\}$ to the end, then pick batch $\{o_3, o_4\} = \{s^{\text{left}}, s^{\text{right}}\}$, at a total cost of $2T$. The second is to turn back and complete a modified batch $S_1 = \{o_1, o_3\} = \{s^{\text{left}}, s^{\text{left}}\}$, then pick $S_2 = \{o_2, o_4\} = \{s^{\text{right}}, s^{\text{right}}\}$, at a total cost of $r_4 + \frac{3}{4}T - \frac{\delta}{2k-2} + \frac{T}{2} = 2T - 2 \cdot \frac{\delta}{2k-2}$. *Reopt* decides for the second option.

At time $r_5 = r_6 = \frac{5}{4}T - 3 \cdot \frac{\delta}{2k-2}$, an order s^{left} and an order s^{right} arrive, and the picker is at a distance of $\frac{\delta}{2k-2}$ above s^{left} with o_1 placed in her cart. *Reopt* has two choices. The first is to continue picking $S_1 = \{o_1, o_3\} = \{s^{\text{left}}, s^{\text{left}}\}$ and then $S_2 = \{o_2, o_4\} = \{s^{\text{right}}, s^{\text{right}}\}$ and $S_3 = \{o_5, o_6\} = \{s^{\text{left}}, s^{\text{right}}\}$, at a total cost of $2T - 2 \cdot \frac{\delta}{2k-2} + T = 3T - 2 \cdot \frac{\delta}{2k-2}$. The second is to change the first batch again to $S_1 = \{o_1, o_2\} = \{s^{\text{left}}, s^{\text{right}}\}$, then pick $S_2 = \{o_3, o_5\} = \{s^{\text{left}}, s^{\text{left}}\}$ and $S_3 = \{o_4, o_6\} = \{s^{\text{right}}, s^{\text{right}}\}$, at a total cost of $r_6 + \frac{3}{4}T - \frac{\delta}{2k-2} + 2 \cdot \frac{T}{2} = 3T - 4 \cdot \frac{\delta}{2k-2}$. *Reopt* decides for the second option.

In general, at time $r_{2p-1} = r_{2p} = \frac{2p-1}{4}T - (2p-3) \cdot \frac{\delta}{2k-2}$ for $p \in \{2, \dots, k\}$, an order s^{left} and an order s^{right} arrive. If p is uneven, the picker is at distance $\frac{\delta}{2k-2}$ above s^{left} planning to pick $\frac{p-1}{2}$ batches of type $\{s^{\text{left}}, s^{\text{left}}\}$ first, then $\frac{p-1}{2}$ batches of type $\{s^{\text{right}}, s^{\text{right}}\}$. With the two recently arrived orders, *Reopt* decides to pick batch $S_1 = \{o_1, o_2\} = \{s^{\text{left}}, s^{\text{right}}\}$ instead, then $\frac{p-1}{2}$ batches of type $\{s^{\text{left}}, s^{\text{left}}\}$ and $\frac{p-1}{2}$ batches of type $\{s^{\text{right}}, s^{\text{right}}\}$. So, the picker turns around towards s^{right} . Similarly, if p is even, the picker is at distance $\frac{\delta}{2k-2}$ above s^{right} planning to pick $S_1 = \{o_1, o_2\} = \{s^{\text{left}}, s^{\text{right}}\}$ first, then $\frac{p}{2} - 1$ batches of type $\{s^{\text{left}}, s^{\text{left}}\}$, and then $\frac{p}{2} - 1$ batches of type $\{s^{\text{right}}, s^{\text{right}}\}$. With the two recently arrived orders, *Reopt* decides to pick $\frac{p}{2}$ batches of type $\{s^{\text{left}}, s^{\text{left}}\}$ first, then $\frac{p}{2}$ batches of type $\{s^{\text{right}}, s^{\text{right}}\}$. So, the picker turns around towards s^{left} .

If k is uneven, $\text{Reopt}(I^{\text{worst}}) = kT - \delta$, which equals the release time of the last order $r_{2k} = \frac{2k-1}{4}T - (2k-3) \cdot \frac{\delta}{2k-2}$ plus the time $(\frac{3}{4}T - \frac{\delta}{2k-2})$ to reach s^{right} and complete the first batch plus $(\frac{k-1}{2} \cdot 2 \cdot \frac{1}{2}T)$ to pick $\frac{k-1}{2}$ batches of type $\{s^{\text{left}}, s^{\text{left}}\}$ and $\frac{k-1}{2}$ batches

of type $\{s^{\text{right}}, s^{\text{right}}\}$. In contrast, *CIOPT* picks $S_1 = \{o_1, o_2\}$, then $\frac{k-1}{2}$ batches of type $\{s^{\text{left}}, s^{\text{left}}\}$ and $\frac{k-1}{2}$ batches of type $\{s^{\text{right}}, s^{\text{right}}\}$ such that $\text{CIOPT}(I(k)) = T + (k-1)\frac{T}{2} = \frac{k}{2}T + \frac{T}{2}$. This implies that

$$\lim_{k \rightarrow \infty} \frac{\text{Reopt}(I(k))}{\text{CIOPT}(I(k))} = 2 \quad (26)$$

By Lemma 5 in Electronic Companion D, $\sigma_{\text{asympt}}(\text{Reopt}) \geq 2$ follows immediately for *Pcart*.

For $c > 2$, we can place multiple single-item orders at positions s^{left} and s^{right} . Note that in the example above, *Reopt* faces ties. Initially, there are two alternative plans with the same cost — picking the orders together as $S_1 = \{o_1, o_2\}$ in a cyclic tour or picking them separately as $S_1 = \{o_1\}$ and $S_2 = \{o_2\}$. In the second plan, the example does not work. We can avoid ties by considering $o_1 = \{s^{\text{left}}, s'\}$, where s' is placed just above s^{left} , which makes *Reopt* to prefer picking both orders together. $\square$

3.4 | Performance of *Reopt* in Instances of All Sizes

In any OOBSP instance, *Reopt* never exceeds more than 4 times the complete-information optimum in *Pcart* and *Robot* and no more than 2.5 times in *Pcart-N*, respectively (see Proposition 4). However, strict competitive ratios of *any* deterministic online algorithm are at least 2, 1.64, and 2 in *Pcart-N*, *Pcart*, and *Robot*, respectively, by Proposition 5. In other words, the performance of *Reopt* cannot be improved much across instances of all sizes.

Observe that the upper bound 2.5 on the strict competitive ratio for *Reopt* in *Pcart-N* is almost tight, as we can construct instances I with $\frac{\text{Reopt}(I)}{\text{CIOPT}(I)} = 2.5 - \bar{\epsilon}$ for an arbitrary small $\bar{\epsilon} > 0$ (see Example 1).

Proposition 4. *The following upper bounds hold for the strict competitive ratio:*

$$\text{In } Pcart - N : \quad \sigma(\text{Reopt}) \leq 2.5 \quad (27)$$

$$\text{In } Pcart \text{ and } Robot : \quad \sigma(\text{Reopt}) \leq 4 \quad (28)$$

Proof of (27). We are in system *Pcart-N* and consider two cases $\square$

In the first case, the picker in *Reopt* is idle at the depot at time r_n . Then, by Lemma 4:

$$\text{Reopt}(I) \leq r_n + \text{CIOPT}(I^{r=0}) \leq 2 \cdot \text{CIOPT}(I) \quad (29)$$

In the other case, at r_n , the picker in *Reopt* is collecting batch $\tilde{S}$, for which she left the depot at time $t(\tilde{S})$. Let $\bar{j}$ be the index of the first order that arrives *after* time $t(\tilde{S})$. Let $I_{\geq \bar{j}}$ denote an instance which contains the subset of orders $\{o_i \mid i \geq \bar{j}\}$ and assumes these orders are *instantly available*. We denote by $\tilde{J}$ an instance that contains all orders in the queue of *Reopt* at time $t(\tilde{S})$, and that assumes all these orders are *instantly available*. By definition, at time $t(\tilde{S})$, *Reopt*'s plan to collect the orders of $\tilde{J}$ is *optimal* given the current information. Therefore, the solution of *Reopt* is not worse than the following feasible policy: At $t(\tilde{S})$, complete first $\tilde{J}$ in time $\text{CIOPT}(\tilde{J})$. After that, the picker is at the depot and

collects the remaining orders in at most $CIOPT(I_{\geq \bar{j}})$ time. We denote $d_{\max} = \max\{d(l_d, s_j^i) \mid s_j^i \in o_{\bar{j}} \cup o_{\bar{j}+1} \cup \dots \cup o_n\}$, the maximal distance from the depot to an item in $I_{\geq \bar{j}}$. Altogether:

$$Reopt(I) \leq t(\tilde{S}) + CIOPT(\tilde{J}) + CIOPT(I_{\geq \bar{j}}) \quad (30)$$

$$\leq r_j + CIOPT(I_{\geq \bar{j}}) + CIOPT(I) \quad (31)$$

$$\leq CIOPT(I) + d_{\max} + CIOPT(I) \quad (32)$$

$$\leq 2.5 \cdot CIOPT(I) \quad (33)$$

Line (31) follows by Lemma 4. In $CIOPT$, the picker can move towards the first picking location prior to the arrival of the corresponding order, thus (32) ensues from Equation (31): $CIOPT(I) + d_{\max} \geq r_j + CIOPT(I_{\geq \bar{j}})$. Finally, since $CIOPT(I) \geq 2 \cdot d_{\max}$, (33) follows.

The logic above does not hold for $Pcart-Int$ and $Robot$, because the initiated batch $\tilde{S}$ can be changed at each order arrival after $t(\tilde{S})$, possibly leading to a worse solution.

Proof of (28). Consider $Pcart$ and recall Lemma 2: At time r_n , when the last order arrives, $Reopt$ reoptimizes having the complete information and requires not more time than the following simple policy: (1) Complete the currently open batch and move to the depot, then (2) use $CIOPT(I(n)^{r=0})$ to collect the remaining orders. For (1), the picker requires at most $r_n + CIOPT(I(n)^{r=0})$, because she can walk from the current position all the way back to the depot in r_n and then follow $CIOPT$ for picking the remaining items of the currently open batch (and skipping the rest). Thus, for (1) together with (2), by Lemma 4:

$$\begin{aligned} Reopt(I(n)) &\leq 2 \cdot r_n + 2 \cdot CIOPT(I(n)^{r=0}) \\ &\leq 4 \cdot CIOPT(I(n)) \end{aligned} \quad (34)$$

The proof for $Robot$ proceeds along the same lines. Observe, however, that at the start of step (2), the picker may be away from the depot. However, she can follow $CIOPT$ in the *reverse* direction to collect the remaining items. $\square$

Figure 6 provides an unfortunate instance I^{unf} for $Pcart-N$ with $c = 2$, such that $\frac{Reopt(I^{unf})}{CIOPT(I^{unf})} \approx 2.5$.

Example 1. In instance I^{unf} of Figure 6, $Reopt$ receives two orders at time $r_1 = r_2 = W + L$: o_1 with only one item s^{close} at

the closest picking location to the depot, $d(l_s, s^{close}) = \epsilon$, and o_2 with two far away picking locations that can be visited best in a cyclic tour of length $2(W + L)$ from the depot. $Reopt$ forms batch $s_1 = \{o_1, o_2\}$, and as soon as the picker leaves the depot (at time $r_3 = W + L + \delta, \delta \rightarrow 0$), a third order $o_3 = o_2$ arrives. Although o_2 and o_3 are perfectly matching, $Reopt$ in $Pcart - N$ cannot modify the commenced batch S_1 , and completes the operation with $S_2 = \{o_3\}$ at time $Reopt(I^{unf}) = r_1 + 2(W + L) + 2(W + L) = 5(W + L)$. On the other hand, $CIOPT$ picks the first item of o_2 and o_3 (in s^{right}) at time $W + L + \delta$ and completes its first batch $\{o_2, o_3\}$ at time $2(L + W) + \epsilon + \delta$. After picking o_1 separately, $CIOPT(I^{unf}) = 2(L + W) + 3\epsilon + \delta$. Since $\delta \rightarrow 0$ and ϵ is the distance from the depot to the closest picking location, we can construct instances with arbitrarily small $(3\epsilon + \delta) > 0$.

We now denote by ALG an arbitrary deterministic online algorithm for OOBSP in each of the studied systems ($Pcart-N$, $Pcart$, and $Robot$). An algorithm is *deterministic*, if for every fixed instance, it always generates the same result when run repeatedly. We differentiate deterministic algorithms from probabilistic algorithms, which may choose their actions randomly. Note that ALG in $Pcart-N$ must be an arbitrary *non-interventionist* algorithm that can update the current picking plan only when the picker is at the depot.

Proposition 5. For every deterministic online algorithm ALG for OOBSP, the strict competitive ratio cannot be better than the following:

$$\text{In } Pcart - N \text{ and } Robot : \quad \sigma(ALG) \geq 2 \quad (35)$$

$$\text{In } Pcart : \quad \sigma(ALG) \geq 1.64 \quad (36)$$

Note that we require $c \geq 2$ for $Pcart-N$ and $Pcart$.

Proof. The lower bounds for $Pcart$ and $Robot$ are based on Theorems 3.1 and 3.3 of Ausiello et al. [30], respectively. In both cases Ausiello et al. [30], construct unfortunate instances for the online TSP *on the real line*. These instances can be transformed to the warehouse environment. The proof requires nontrivial extensions, which are given in the Electronic Companion F. Indeed, in the warehouse, the movements of the picker are not limited to a real line, and the batching capacity should be respected.

To prove the lower bound for any ALG in $Pcart-N$, consider the instance in Figure 7. Recall that the warehouse width is W

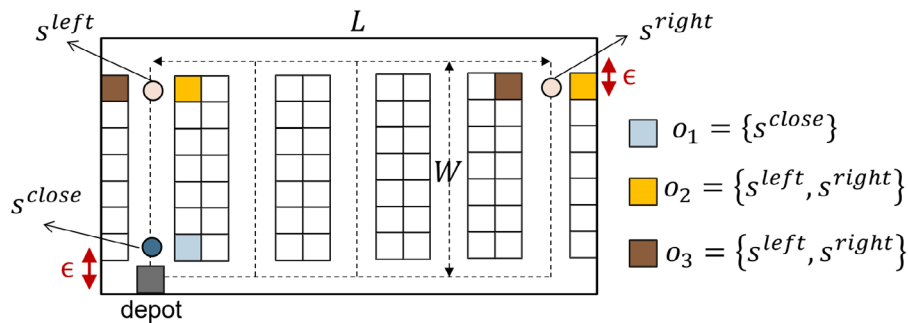


FIGURE 6 | $Pcart-N$: Unfortunate small instance I^{unf} for $Reopt$. $c = 2, r_1 = r_2 = L + W$, and $r_3 = L + W + \delta$ with $\delta \rightarrow 0$. $Reopt$ myopically collects poorly matching orders o_1 and o_2 together.

and the picker moves at the unit speed. Initially, at $r_1 = 0$, order $o_1 = \{s_1^1\}$ with $d(l_d, s_1^1) = \frac{1}{2}W$ is available. Let *ALG* decide to dispatch the picker to collect the single-order batch $\{o_1\}$ at time t . In the following, we will construct an unfortunate instance for each possible value of t .

If $t \geq W$, no more orders arrive. *ALG* needs $(t + W) \geq 2W$ time to collect o_1 in a return-trip from the depot, while $CIOPT(I) = W$ for the resulting instance I .

If $\frac{1}{2}W \leq t < W$, then a second order $o_2 = \{s_2^1\}$ with $d(l_d, s_2^1) = t$ arrives at time $r_2 = t$ (but after the picker dispatched). Then, in the resulting instance I , batch $\{o_1, o_2\}$ can be collected in time $CIOPT(I) = 2t$, while *ALG* collects o_1 and o_2 in two separate batches with $ALG^{Pcart-N}(I) = t + W + 2t > 4t$.

If $t < \frac{1}{2}W$, then a second order $o_2 = \{s_2^1\}$ with $d(l_d, s_2^1) = \frac{1}{2}W$ arrives at time $r_2 = t$ (but after the picker dispatched). For the resulting instance I , it is possible to collect both orders in the single batch $\{o_1, o_2\}$ in time $CIOPT(I) = W$, while $ALG(I) = t + W + W \geq 2W$, because it picks o_1 and o_2 in separate batches.

In all the cases above, the ratio $\frac{ALG}{CIOPT(I)} \geq 2$ for the respective instance I in *Pcart-N*.

Observe that in the unfortunate examples for *Pcart-N* and *Pcart*, $c \geq 2$. If $c = 1$, a weaker lower bound of 1.5 can be

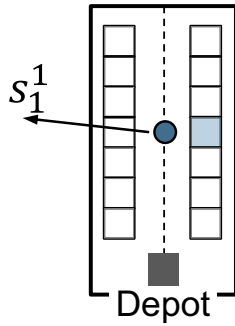


FIGURE 7 | *Pcart-N*: Unfortunate small instance for all algorithms. No action suits all possible futures well: (a) no further orders arrive, (b) further orders arrive.

straightforwardly derived in both cases from the unfortunate instance created for OOSRP in *Robot* in the Electronic Companion F. $\square$

4 | Computational Experiments

The analysis of Section 3.2 showed that *Reopt* may converge to an optimal policy with an increasing instance size. In this section, we compare the results of *Reopt* to optimality (measured by *CIOPT*) experimentally and investigate the gap to optimality in small instances.

Note the computational challenges associated with these experiments. To make representative measurements across diverse warehouse settings, we create 2600 OOSRP* instances and solve more than 2380 of those to optimality. The quantity of items that need to be picked in the created instances ranges between 4 and 50, and OOSRP*, which combines batching and routing, is NP-hard in the strong sense (cf. [40]). Given the scale of the endeavor, we limit our studies to the interventionist versions of *Reopt*, that is, systems *Pcart* and *Robot*. Section 4.1 explains the data generation, and Section 4.2 reports on experiments.

4.1 | Data Sets

We randomly generate instances grouped in 10 settings (see Table 2) that feature different warehouse environments. Each setting comprises 10 instances of each order size $n \in \{3, 4, \dots, 15\}$. The procedure is repeated for two picking systems—*Pcart* and *Robot*. Thus, in total, 2 600 instances (2 systems $\times$ 10 settings $\times$ 13 sizes $\times$ 10) are created.

In the *Base setting*(*Base*), the picker's speed equals 0.8 m/s (cf. [28]) and it takes 10 s to collect one item at each picking location. The warehouse consists of 10 aisles and 3 cross-aisles, each of which has a width of 3 m. One single shelf row marks the left and right outer borders of the storage area, respectively, and neighboring aisles are separated by two shelf rows placed alongside each other (cf. Figure 2). The shelves measure 3 m in width

TABLE 2 | Overview of the 10 settings of the data set.

Changed factor		Settings	
Number of aisles	—	Base (10 aisles)	Large warehouse (20 aisles)
Number of cross-aisles	Less cross-aisles (2 cross-aisles)	Base (3 cross-aisles)	—
Storage policy	—	Base (uniform)	Class-based dispersion (class-based)
Maximal order size	Small orders (2 SKUs)	Base (4 SKUs)	Large orders (8 SKUs)
Batching capacity	Small batches ($c = 1$)	Base ($c = 2$)	Large batches ($c = 4$)
Arrival rate	Small rate ($r^{Pcart} = 70, r^{Robot} = 90$)	Base ($r^{Pcart} = 90, r^{Robot} = 110$)	Large rate ($r^{Pcart} = 110, r^{Robot} = 130$)

and 1 m in depth. One aisle contains 20 shelves. Thus, the warehouse has a length of 50 m (10 aisles $\times$ 3 m + (1 + 9 $\cdot$ 2 + 1) shelf rows $\times$ 1 m) and a width of 69 m (20 shelves $\times$ 3 m + 3 cross-aisles $\times$ 3 m). Each shelf has slots for three distinct *stock keeping units* (SKUs), that is, distinguishable product items, which results in a granularity of 1 m and provides 1200 slots for SKUs (20 shelves per aisle $\times$ 3 $\times$ 20 shelf rows) in the warehouse in total. The depot is positioned at the lower-left corner of the warehouse. The batch capacity is $c = 2$. The number of items $k(j)$ in each order o_j is uniformly distributed in set the $\{1, 2, \dots, \bar{k}\}$, $\bar{k} = 4$, which reflects a common scenario in e-commerce, where the typical order contains fewer than two items on average (cf. [12, 44]). The storage locations of ordered items are uniformly distributed among all storage locations (*uniform dispersion*). The orders arrive according to a homogeneous Poisson process with an arrival rate r . In preliminary experiments, we investigated different arrival rates r for shifts of 4h, increasing the rates in steps of 10 orders/shift, and found that optimal arrival rates in the *Base* setting, at which the system remains stable, are $r^{\text{Pcart}} = 90$ orders/shift and $r^{\text{Robot}} = 110$ orders/shift for *Pcart* and *Robot*, respectively. So, we set the arrival rates to these numbers. For *Pcart*, the 130 instances in *Base*, which form 13 groups of 10 instances according to their size n , are generated randomly and independently from each other. For better comparability, the 130 instances in *Base* for *Robot* are identical to those in *Pcart*, except for the sequence of arrival times, which was produced using a higher stable rate r^{Robot} . We generate the remaining nine settings by taking the instances of *Base* and changing their specific characteristics in a one-factor-at-a-time design (see Table 2). For better control of the impact of warehouse characteristics, we keep the instances unchanged (including the coordinates of the generated items and the order arrival times) over the different settings, unless explicitly stated otherwise.

We investigate different batch capacities in settings *Smallbatches* and *Largebatches*, where we set $c = 1$ and $c = 4$, respectively (all other instance data remained unchanged). In *Smallrate* and *Largerate*, the arrival rates are set to $r^{\text{Pcart}} = 70$, $r^{\text{Pcart}} = 110$ and $r^{\text{Robot}} = 90$ and $r^{\text{Robot}} = 130$ for *Robot* and *Pcart*, respectively. We do it by generating new arrival times for the instances in *Base*. We investigate warehouse geometries in *Largewarehouse* (20 aisles) and *Lesscrossaisles* (2 cross-aisles). Observe that the lengths and widths of the shelves, aisles, and cross-aisles, as well as the number of shelves per aisle, remain the same, so that the size of the warehouse changes accordingly. In particular, in *Largewarehouse*, the warehouse has a length of 100 m (20 aisles $\times$ 3 m + (1 + 19 $\cdot$ 2 + 1) shelf rows $\times$ 1 m) and a width of 69 m is identical to *Base*; in *Lesscrossaisles*, the warehouse length of 50 m is identical to *Base*, and the width is 66 m (20 shelves $\times$ 3 m + 2 cross-aisles $\times$ 3 m). In *Largewarehouse*, we generate the positions of the ordered items anew. We do not have to do it in *Lesscrossaisles*, since the number of SKU slots remains the same (1 200) and only the middle cross-aisle disappears. In *Classbaseddispersion*, SKUs are stored in the warehouse according to their turnover rates, so that we take instances of *Base* and generate new picking positions. We randomly assign ordered items to three classes: A, B, and C with probabilities of 52%, 36%, and 12%. Items in A refer to high-turnover products, which occupy slots in the first aisle. Items in B occupy aisles 2 to 4. And aisles in C are low-turnover products, which occupy aisles 5 to 10. We randomly assign a storage position to each ordered item in the respective aisles of its

class. Finally, in *Smallorders* and *Largeorders*, the maximal order size is set to $\bar{k} = 2$ and $\bar{k} = 8$, respectively. For this, we have to generate the orders (including their size, positions of the items, and arrival times) anew in the instances of *Base*. The generated instances can be found in the Electronic Companion.

We performed the experiments on the Compute Canada cluster by using 1 CPU with a time and memory limit of at most 1h and 350G, respectively. For each instance I , we compute $CIOPT(I)$ by solving the respective OBSRP* exactly with the dynamic programming approach described in Lorenz et al. [42]. We calculate $Reopt(I)$ by letting the orders arrive dynamically according to their arrival rates and applying exact reoptimization as described in Section 2.2 using the straightforwardly adapted dynamic programming approach of Lorenz et al. [42].

4.2 | Optimality Ratios of *Reopt* in *Pcart* and *Robot*

For each instance I , we compute the *optimality ratio* as $\frac{Reopt(I)}{CIOPT(I)}$. Tables 3 and 4 report worst observed optimality ratios in *Pcart* and *Robot*. The average observed optimality ratios for *Pcart* and *Robot* can be found in Tables S5 and S6 in Electronic Companion G.

According to our experiments, *Reopt*'s results are rather close to optimality even in small-sized instances. For the smallest tested instance sizes (with the number of orders $n < 5$), the average ratio over all settings equaled 1.10 and 1.09, for *Pcart* and *Robot*, respectively; for the largest tested instance sizes ($n > 13$), it decreased to 1.05, for both cart types. The worst observed optimality ratio across *all* the instances equaled 1.34 for *Pcart* and 1.32 for *Robot*; and the respective ratios for larger instances ($n > 13$) decreased and equaled 1.13 and 1.16, respectively.

To statistically validate the monotonic decreasing relationship between the observed optimality ratios and the instance size n , we conducted a Spearman's rank correlation test (see e.g., [45]). For each of the 10 settings, we computed the Spearman's correlation coefficient for the optimality ratios across all solved instances of different instance sizes (130 observations for most settings, fewer for *Largeorders* and *Largebatches*, see Tables 3 and 4). In *Pcart* (*Robot*), for all settings except *Largeorders*, the coefficients ranged from -0.68 to -0.25 (from -0.61 to -0.27) with p -values of less than 0.04 (less than 0.01) indicating a moderate but statistically significant decrease. For *Largeorders*, the Spearman's correlation coefficient was -0.23 (-0.16), and the p -value of 0.07 (0.20) may be explained by the lower number of observations and the limitation in instance size n .

5 | Conclusion

This paper performs an optimality analysis of reoptimization policies for the *Online Order Batching, Sequencing, and Routing Problem* (OBSRP). The focus is on *immediate reoptimization* (*Reopt*), where the current (partial) solution adjusts each time a new order arrives. We consider warehouses employing manual pushcarts and warehouses employing robotic carts. Given that picking instructions in traditional warehouses may only be transmitted at the depot, we additionally examine the *non-interventionist* version of *Reopt* for manual pushcarts. In

TABLE 3 | Worst observed optimality ratios of *Reopt* in *Pcart*.

Setting	The number of orders $n =$												
	3	4	5	6	7	8	9	10	11	12	13	14	15
Base	1.18	1.14	1.16	1.12	1.15	1.09	1.20	1.08	1.14	1.14	1.07	1.10	1.11
Largewarehouse	1.21	1.15	1.15	1.18	1.18	1.12	1.18	1.15	1.11	1.12	1.13	1.12	*1.10
Lesscrossaisles	1.16	1.24	1.18	1.10	1.10	1.14	1.14	1.10	1.12	1.09	1.07	1.13	1.13
Classbaseddispersion	1.20	1.14	1.18	1.12	1.16	1.11	1.14	1.14	1.13	1.11	1.10	1.10	*1.08
Smallorders	1.34	1.33	1.22	1.11	1.16	1.08	1.16	1.08	1.07	1.09	1.23	1.10	1.09
Largeorders	1.18	1.21	1.10	1.08	1.12	*1.17	*1.10	—	—	—	—	—	—
Smallbatches	1.15	1.15	1.08	1.08	1.07	1.06	1.06	1.05	1.05	1.04	1.04	1.04	1.03
Largebatches	1.17	1.17	1.31	1.16	1.17	1.12	1.13	—	—	—	—	—	—
Smallrate	1.14	1.17	1.13	1.11	1.08	1.12	1.05	1.05	1.10	1.15	1.08	1.12	1.09
Largerate	1.19	1.14	1.19	1.21	1.12	1.11	1.13	1.12	1.12	1.14	1.13	1.11	*1.08
Total	1.34	1.33	1.31	1.21	1.18	1.17	1.20	1.15	1.14	1.15	1.23	1.13	1.13

Note: *Results for less than 10 (out of 10) instances. CIOPT did not solve 1 or 2 instances to optimality. —No results reported, since CIOPT solved less than 8 instances to optimality.

TABLE 4 | Worst observed optimality ratios of *Reopt* in *Robot*.

Setting	The number of orders $n =$												
	3	4	5	6	7	8	9	10	11	12	13	14	15
Base	1.18	1.24	1.12	1.10	1.18	1.05	1.12	1.09	1.06	1.09	1.07	1.11	1.08
Largewarehouse	1.32	1.14	1.19	1.24	1.12	1.08	1.12	1.09	1.07	1.08	1.11	1.09	1.10
Lesscrossaisles	1.25	1.25	1.31	1.12	1.16	1.13	1.13	1.10	1.17	1.11	1.14	1.16	1.12
Classbaseddispersion	1.15	1.22	1.10	1.18	1.14	1.18	1.08	1.09	1.08	1.04	1.08	1.10	1.10
Smallorders	1.19	1.29	1.17	1.11	1.10	1.08	1.13	1.07	1.08	1.06	1.06	1.12	1.07
Largeorders	1.12	1.11	1.22	1.07	1.10	*1.12	*1.10	—	—	—	—	—	—
Smallbatches	1.22	1.24	1.19	1.06	1.15	1.08	1.14	1.11	1.09	1.07	1.10	1.06	1.06
Largebatches	1.18	1.24	1.15	1.09	1.11	1.11	1.15	*1.12	—	—	—	—	—
Smallrate	1.22	1.15	1.11	1.08	1.07	1.09	1.09	1.10	1.10	1.09	1.06	1.10	1.08
Largerate	1.25	1.12	1.23	1.08	1.17	1.09	1.15	1.10	1.09	1.09	1.13	1.11	*1.08
Total	1.32	1.29	1.31	1.24	1.18	1.18	1.15	1.12	1.17	1.11	1.14	1.16	1.12

Note: *Results for less than 10 (out of 10) instances. CIOPT did not solve 1 or 2 instances to optimality. —No results reported, since CIOPT solved less than 8 instances to optimality.

total, we investigate three systems, each featuring a specific reoptimization policy and cart technology: *Robot* for interventionist *Reopt* and robotic carts, *Pcart* for interventionist *Reopt* and manual pushcarts, as well as *Pcart-N* for non-interventionist *Reopt* and manual pushcarts.

OOBSRP can be interpreted as a nontrivial combination of the *online batching problem* and the *online traveling salesman problem*. To the best of our knowledge, there is currently no clarity on the performance gaps to optimality for *any* OOBSRP policy. *Reopt* is chosen for analysis due to its prominence in warehousing literature and its ability to leverage recent algorithmic progress for the offline version of OOBSRP.

Our extended competitive analysis demonstrates the *almost sure asymptotically optimal* performance of *Reopt* under very general stochastic assumptions in all three examined systems—*Robot*,

Pcart, and *Pcart-N*. Our stochastic assumptions can describe a variety of order arrival patterns relevant for practice. On the formal side, they encompass arrival times that are modeled as order statistics or as a homogeneous Poisson process with specific arrival rates.

Even if we drop stochastic assumptions and examine *any* theoretically possible OOBSRP instances, *Reopt* performs well, with no policy improving its result by more than 50% in sufficiently large instances. Anomalies are possible in small OOBSRP instances, which cannot be overcome by *any* deterministic algorithm (see strict competitive ratios in Section 3.4). However, no policy can improve *Reopt*'s result by more than 75% regardless of the instance size.

Having established the almost sure *asymptotic* optimality of *Reopt* analytically under various stochastic scenarios, we

show empirically that *Reopt* remains rather close to the complete-information optimality already in small instances in *Pcart* and *Robot*.

This paper provides insights into ongoing debates regarding the benefits of waiting and anticipation for online policies in warehouse operations. Indeed, the performance gaps of *Reopt*, which eschews both waiting and anticipation, serves as a benchmark for assessing the benefits of these concepts (see Section 1.2).

Several questions remain for future research, including the open status of Conjecture 1 and determining *stable rates* λ when order arrival times follow a Poisson process. Further performance analysis is needed for other online algorithms, including simple policies like “first-come-first-served” and S-shape routing. An optimal online policy is still unknown, especially for small OOB-SRP instances. Exploration of additional OOB-SRP variants and extensions, including multiple pickers, picking location assignment, advanced information on orders, and order due dates, is suggested as avenues for future research. Furthermore, the analysis of alternative objectives, particularly those related to the service level (e.g., tardiness), remains an outstanding research area.

Acknowledgments

We would like to extend our gratitude to the Government of Quebec (Ministère des Relations internationales et de la Francophonie) and BayFor for their support. We also acknowledge Compute Canada for providing the computational resources necessary for this research. Open Access funding enabled and organized by Projekt DEAL.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

1. T.-L. Chen, C.-Y. Cheng, Y.-Y. Chen, and L.-K. Chan, “An Efficient Hybrid Algorithm for Integrated Order Batching, Sequencing and Routing Problem,” *International Journal of Production Economics* 159 (2015): 158–167.
2. G. Marchet, M. Melacini, and S. Perotti, “Investigating Order Picking System Adoption: A Case-Study-Based Approach,” *International Journal of Logistics Research and Applications* 18, no. 1 (2015): 82–98.
3. M. Napolitano, “Warehouse/DC Operations Survey: Mixed Signals,” *Logistics Management (Highlands Ranch, Colo.: 2002)* 51, no. 11 (2012): 51–55.
4. S. Vanheusden, T. van Gils, K. Ramaekers, T. Cornelissens, and A. Caris, “Practical Factors in Order Picking Planning: State-Of-The-Art Classification and Review,” *International Journal of Production Research* 61, no. 6 (2023): 2032–2056.
5. M. Löffler, N. Boysen, and M. Schneider, “Human-Robot Cooperation: Coordinating Autonomous Mobile Robots and Human Order Pickers,” *Transportation Science* 57, no. 4 (2023): 839–1114.
6. M. Schiffer, N. Boysen, P. S. Klein, G. Laporte, and M. Pavone, “Optimal Picking Policies in e-Commerce Warehouses,” *Management Science* 68, no. 10 (2022): 7497–7517.
7. C. A. Valle, J. E. Beasley, and A. S. Da Cunha, “Optimally Solving the Joint Order Batching and Picker Routing Problem,” *European Journal of Operational Research* 262, no. 3 (2017): 817–834.
8. D. Hwang and P. Jaillet, “Online Scheduling With Multi-State Machines,” *Networks* 71, no. 3 (2018): 209–251.
9. P. Jaillet and M. R. Wagner, “Generalized Online Routing: New Competitive Ratios, Resource Augmentation, and Asymptotic Analyses,” *Operations Research* 56, no. 3 (2008): 745–757.
10. A. Borodin and R. El-Yaniv, *Online Computation and Competitive Analysis* (Cambridge University Press, 1998).
11. E. G. Pardo, S. Gil-Borrás, A. Alonso-Ayuso, and A. Duarte, “Order Batching Problems: Taxonomy and Literature Review,” *European Journal of Operational Research* 313, no. 1 (2024): 1–24.
12. N. Boysen, R. De Koster, and F. Weidinger, “Warehousing in the e-Commerce Era: A Survey,” *European Journal of Operational Research* 277, no. 2 (2019): 396–411.
13. Y. Li, R. Zhang, and D. Jiang, “Order-Picking Efficiency in e-Commerce Warehouses: A Literature Review,” *Journal of Theoretical and Applied Electronic Commerce Research* 17, no. 4 (2022): 1812–1830.
14. T. Van Gils, A. Caris, K. Ramaekers, and K. Braekers, “Formulating and Solving the Integrated Batching, Routing, and Picker Scheduling Problem in a Real-Life Spare Parts Warehouse,” *European Journal of Operational Research* 277, no. 3 (2019): 814–830.
15. T. Le-Duc and R. M. De Koster, “Travel Time Estimation and Order Batching in a 2-Block Warehouse,” *European Journal of Operational Research* 176, no. 1 (2007): 374–388.
16. I. Van Nieuwenhuyse and R. B. De Koster, “Evaluating Order Throughput Time in 2-Block Warehouses With Time Window Batching,” *International Journal of Production Economics* 121, no. 2 (2009): 654–664.
17. M. Yu and R. B. De Koster, “The Impact of Order Batching and Picking Area Zoning on Order Picking System Performance,” *European Journal of Operational Research* 198, no. 2 (2009): 480–490.
18. M. Yu and R. De Koster, “Performance Approximation and Design of Pick-And-Pass Order Picking Systems,” *IEEE Transactions* 40, no. 11 (2008): 1054–1069.
19. M. Schleyer and K. Gue, “Throughput Time Distribution Analysis for a One-Block Warehouse,” *Transportation Research Part E: Logistics and Transportation Review* 48, no. 3 (2012): 652–666.
20. Y. Bukchin, E. Khmelnitsky, and P. Yakuel, “Optimizing a Dynamic Order-Picking Process,” *European Journal of Operational Research* 219, no. 2 (2012): 335–346.
21. B. Cals, Y. Zhang, R. Dijkman, and C. van Dorst, “Solving the Online Batching Problem Using Deep Reinforcement Learning,” *Computers & Industrial Engineering* 156 (2021): 107221.
22. R. D’Haen, K. Braekers, and K. Ramaekers, “Integrated Scheduling of Order Picking Operations Under Dynamic Order Arrivals,” *International Journal of Production Research* 61, no. 10 (2023): 3205–3226.
23. J. Zhang, X. Wang, and K. Huang, “Integrated On-Line Scheduling of Order Batching and Delivery Under B2C e-Commerce,” *Computers & Industrial Engineering* 94 (2016): 280–289.
24. J. Zhang, X. Wang, F. T. Chan, and J. Ruan, “On-Line Order Batching and Sequencing Problem With Multiple Pickers: A Hybrid Rule-Based Algorithm,” *Applied Mathematical Modelling* 45 (2017): 271–284.
25. W. Lu, D. McFarlane, V. Giannikas, and Q. Zhang, “An Algorithm for Dynamic Order-Picking in Warehouse Operations,” *European Journal of Operational Research* 248, no. 1 (2016): 107–122.
26. V. Giannikas, W. Lu, B. Robertson, and D. McFarlane, “An Interventionist Strategy for Warehouse Order Picking: Evidence From Two Case Studies,” *International Journal of Production Economics* 189 (2017): 63–76.
27. H. Dauod and D. Won, “Real-Time Order Picking Planning Framework for Warehouses and Distribution Centres,” *International Journal of Production Research* 60, no. 18 (2022): 5468–5487.

28. S. Henn, "Algorithms for On-Line Order Batching in an Order Picking Warehouse," *Computers & Operations Research* 39, no. 11 (2012): 2549–2563.
29. M. Alipour, Y. Z. Mehrjerdri, and A. Mostafaeipour, "A Rule-Based Heuristic Algorithm for On-Line Order Batching and Scheduling in an Order Picking Warehouse With Multiple Pickers," *RAIRO - Operations Research* 54, no. 1 (2020): 101–107.
30. G. Ausiello, E. Feuerstein, S. Leonardi, L. Stougie, and M. Talamo, "Algorithms for the On-Line Travelling Salesman," *Algorithmica* 29 (2001): 560–581.
31. G. Zhang, X. Cai, and C. Wong, "On-Line Algorithms for Minimizing Makespan on Batch Processing Machines," *Naval Research Logistics* 48, no. 3 (2001): 241–258.
32. Z. Liu and W. Yu, "Scheduling One Batch Processor Subject to Job Release Dates," *Discrete Applied Mathematics* 105, no. 1–3 (2000): 129–136.
33. N. Ascheuer, S. O. Krumke, and J. Rambau, "Online Dial-a-Ride Problems: Minimizing the Completion Time," in *STACS 2000: 17th Annual Symposium on Theoretical Aspects of Computer Science Lille, France, February 17–19, 2000 Proceedings 17* (Springer, 2000), 639–650.
34. E. Angelelli, M. G. Speranza, and M. W. P. Savelsbergh, "Competitive Analysis for Dynamic Multiperiod Uncapacitated Routing Problems," *Networks* 49, no. 4 (2007): 308–317.
35. P. Jaillet and X. Lu, "Online Traveling Salesman Problems With Service Flexibility," *Networks* 58, no. 2 (2011): 137–146.
36. P. Jaillet and X. Lu, "Online Traveling Salesman Problems With Rejection Options," *Networks* 64, no. 2 (2014): 84–95.
37. D. J. Bertsimas and G. van Ryzin, "Stochastic and Dynamic Vehicle Routing in the Euclidean Plane With Multiple Capacitated Vehicles," *Operations Research* 41 (1993): 60–76.
38. C.-M. Chen, Y. Gong, R. de Koster, and J. van Nunen, "A Flexible Evaluative Framework for Order Picking Systems," *Production and Operations Management* 19 (2010): 70–82.
39. M. W. Ulmer, *Approximate Dynamic Programming for Dynamic Vehicle Routing*, vol. 61 (Springer, 2017).
40. M. Löffler, N. Boysen, and M. Schneider, "Picker Routing in AGV-Assisted Order Picking Systems," *INFORMS Journal on Computing* 34, no. 1 (2022): 440–462.
41. I. Žulj, H. Salewski, D. Goeke, and M. Schneider, "Order Batching and Batch Sequencing in an AMR-Assisted Picker-To-Parts System," *European Journal of Operational Research* 298, no. 1 (2022): 182–201.
42. C. Lorenz, A. Otto, and M. Gendreau, "On Picking Operations in e-Commerce Warehouses: Insights From the Complete-Information Counterpart," (2024), <https://arxiv.org/abs/2410.14316>.
43. A. Fiat and G. J. Woeginger, *Online Algorithms*, vol. 1442 (Springer, 1998).
44. L. Xie, H. Li, and L. Luttmann, "Formulating and Solving Integrated Order Batching and Routing in Multi-Depot AGV-Assisted Mixed-Shelves Warehouses," *European Journal of Operational Research* 307, no. 2 (2023): 713–730.
45. G. Corder and D. Foreman, *Comparing Variables of Ordinal or Dichotomous Scales: Spearman Rank-Order, Point-Biserial, and Biserial Correlations* (John Wiley & Sons, Ltd, 2009), 122–154.

Supporting Information

Additional supporting information can be found online in the Supporting Information section. **Online Appendix:** Supporting Information.