Dynamic Store Fulfillment with Collaborative Robots and In-Store Customers

Joyjit Bhowmick, Jennifer Pazour
Industrial and Systems Engineering, Rensselaer Polytechnic Institute, USA

Iman Dayarian
Culverhouse College of Business, University of Alabama, USA

Omnichannel services, such as buy-online-pickup-in-store, curbside pickup, and ship-from-store, have shifted the order-picking tasks that used to be completed by in-store customers doing their own shopping to the responsibility of retailers. To fulfill these orders, many retailers have deployed a store fulfillment strategy, where online orders are picked from inventory in brick-and-mortar stores. As store fulfillment is currently a labor-intensive operation, we propose an innovative approach that relies on the assistance of in-store customers for item extraction from the store shelves and a fleet of collaborative robots to collect and transport them to a designated station. While collaborative robots are manageable by the store, the arrival of in-store customers who are willing to assist a collaborative robot at a given location in the store is out of the store's control, and therefore, uncertain. We model the stochastic order-picking problem with uncertain synchronization times of in-store customers and collaborative robots as a Markov Decision Process to determine how a retailer should dynamically assign tasks to a set of collaborative robots and dedicated pickers. We develop a heuristic solution framework that generates a set of initial assignments and routes for picking resources and dynamically updates them as the actual synchronization times between collaborative robots and in-store customers unfold. We analyze multiple strategies to generate the initial set of task assignments and routes as well as update such decisions based on the system state. We test our proposed approaches using actual online grocery data. Computational results illustrate the potential for collaborative robots and in-store customers to achieve equivalent pick rates as systems with only dedicated pickers. Lastly, our solution approach is capable of generating high-quality solutions at a pace suitable for practical settings.

Key words: Dynamic Decision making, Order Picking, Retail, Markov Decision Process, Collaborative Robots

1. Introduction

The retail industry, in general, and the grocery sector in particular, have seen the popularity of e-commerce rise both for curbside pickup and home delivery (Mayumi Brewster 2022). During the stay-at-home orders associated with the COVID-19 pandemic, about 40% of Americans (or approximately 39.5 million households) in 2020 tried an online grocery service, resulting in a year-over-year increase of 193% (Brick Meets Clicks 2020). The demand for online grocery services is still at higher than the pre-pandemic levels (Shen et al. 2022) and is expected to continue growing in the coming years (Aull et al. 2022).

Store fulfillment, in which the same inventory on store shelves is used for both online and in-store customers, is a popular way that retailers are meeting the growing demand for online grocery services (Eriksson et al. 2022). While there are advantages to using store fulfillment methods, including pooling demand from both online and in-store channels, being in closer proximity to customers, and utilizing a retailer's existing facilities and supply chain infrastructure, the industry is currently facing significant challenges (Brittain Ladd 2023). A store on average achieves a profit margin of $4.40 on a typical $100 basket of groceries when the in-store customer does their own shopping and these margins become negative when the store is responsible for picking the items (McKinsey 2022). Thus, even without last mile delivery responsibilities, and even after charging a service fee of $4-7, many stores incur negative margins for their curbside pickup services (Repko 2020). Another issue is the current and projected labor shortages (Stephanie Ferguson 2023, Roy Maurer 2023), as well as rising labor costs (U.S. Bureau of Labor Statistics 2023). As a result, more companies are deploying automated resources to meet demand requirements (Begley et al. 2019). However, in terms of pick rates, reliability, and product range, robotic piece-level extraction still lags behind human benchmarks (Christensen et al. 2021, Yu et al. 2020, Correll et al. 2016). Instead, a specific type of collaborative robots (cobots) that work side by side with humans (Franklin et al. 2020), known as autonomous mobile robots (AMRs) has become a promising option for order fulfillment tasks in distribution centers. As shown in Figure 1(b), these cobots can autonomously move around a facility and wait at a pick location, where human employees then extract requested items from shelves and put them in the cobot's order totes (Locus Robotics 2023).

In this work, we consider the potential to deploy AMRs in a retail store for store fulfillment. In the proposed store fulfillment policy, participating in-store shoppers assist the cobots by picking items off store shelves and placing them in a cobot's tote. Specifically, a set of cobots would travel the shopping aisles, stopping and waiting in front of store shelves, each displaying on its monitor a request to pick specific item(s). When a participating in-store customer travels near the cobot, that person can pick the requested items from store shelves, scan them on the cobot's scanner, and drop them in the designated tote on the cobot. Immediately, the cobot's monitor will show a QR code that a customer can scan to get compensated via store credit. To meet online order deadlines, stores would also employ dedicated pickers to also be conducting order picking tasks in the retail store. These dedicated pickers would have their own order picking carts and do not need to interface with the cobots nor the in-store customers.

As in-store customers will have traveled to the store and are traversing the store's aisles to conduct their own shopping, by utilizing in-store customers, in conjunction with cobots, this approach has the potential to reduce the marginal costs of store fulfillment operations. Yet, in-store customers are a unique picking resource, not controlled by the store. Given a store must meet service commitments on online orders, a challenge with this policy is that there is uncertainty in in-store customer arrivals at places where a cobot is waiting for help. Due to the random nature of customer arrivals to different parts of the store, the time a cobot in front of a store shelf must wait until a participating in-store customer arrives (referred to as the synchronization time) is stochastic. This leads to open questions about how best should a store deploy cobots and dedicated pickers in the face of the uncertainty of participating in-store customer arrivals to synchronize with cobots. This environment requires the store to make concurrent assignment and routing decisions for the set of cobots and the set of dedicated pickers that balance the need to meet demand-side service commitments with limited resources in an uncertain retail environment. Furthermore, to maintain a high level of service, a cobot might need to move to another location in the store instead of waiting for a participating in-store customer to arrive at the current location. Thus, the central challenge of formulating this problem is that the store does not have prior knowledge on when an in-store customer will arrive to help a cobot. This motivated us to develop a dynamic decision-making approach, i.e., as new information becomes available in the system, the store has an opportunity to update its next decision. The contributions of this work can be summarized as follows:

- We introduce a new store fulfillment concept that deploys dedicated pickers and autonomous mobile robots (AMRs) and utilizes a previously untapped set of resources, in-store customers, to help pick online orders in a retail store environment.
- We formalize the decision-making process of dynamic resource-to-item assignments, as well as sequencing (routing) and abandonment decisions of picking resources in a time-constrained environment as a Markov Decision Process (MDP). Uncertainty arises from in-store customers being an uncontrollable picking resource, and hence, stochastic synchronization times occur with the cobots.
- We develop a heuristic solution framework that allows for exploration of alternative policy designs in terms of initial sequencing decisions, abandonment strategies, and picking assignment reallocation, as well as dynamically updating such decisions as new information is revealed.
- We provide insights into the operational design of the proposed store fulfillment concept through a set of computational experiments based on actual online orders and empirical consumer shopping behavior data, by answering: (1) What is the potential benefit of deploying dynamic vs. static assignment policies? (2) In which situations is there value in a store deploying a cobot abandonment policy? (3) How does the allowed number of reallocation of stopping points impact

picking performance? (4) How are picking tasks distributed among the dedicated pickers and the cobots? (5) How does the participation rate of in-store customers affect such policies? (6) Are our solution approaches computationally tractable for decision-making in practical settings? (7) Which policy should a store utilize? (8) How many cobots are needed to replace one dedicated picker? And lastly, (9) Is the proposed approach economically viable?

2. Literature Review

Order picking with the help of cobots in distribution centers is an emerging research area (Fragapane et al. 2021, Azadeh et al. 2019, Boysen et al. 2019, Jacob et al. 2023, Lorson et al. 2023, Löffler et al. 2023). A recent focus of research has been on how to design operational policies having collaborative robots, coupled with either humans (Azadeh et al. 2020, Ghelichi and Kilaru 2021, Meller et al. 2018, Löffler et al. 2021, 2023, Zou et al. 2019, Yokota 2019, Pasparakis et al. 2021, Winkelhaus et al. 2022, Srinivas and Yu 2022, Fager et al. 2021, Žulj et al. 2022, Zhang et al. 2021, Zhu et al. 2022) or with other robotic resources (Lee and Murray 2019, Wang et al. 2020) to perform order picking tasks. These papers deploy a wide range of methodologies, primarily deterministic integer programming models to decide on routing, order batching, zoning, and sequencing in a warehouse setting. Additionally, existing work has explored tractable heuristic solution approaches, queuing-based models to capture resource congestion impacts on performance metrics, simulation models to assess validity of such policies, Markov Decision Processes to choose between strategies, and physical lab experiments to understand collaborative behaviors. The most closely related paper is a recent one by Löffler et al. (2023) who studies how to route AMRs and human pickers in a distribution center order fulfillment process. Similar to our work, they also consider the need for AMRs and human pickers to synchronize, yet, as both AMRs and human resources are controllable, in contrast to this work, they develop a static, deterministic integer programming model to make coordinated routing decisions, and present heuristic methods to solve the problem. To the best of our knowledge, all of the above referenced papers use cobots for distribution center order fulfillment, where all picking resources are controlled by the warehouse and thus, none captures stochastic picking tasks. While dynamic decision making in a stochastic warehouse environment has been an area of interest (Bukchin et al. 2012, Lu et al. 2016, Gong and De Koster 2008, Han et al. 2022, Azadeh et al. 2020), the sources of uncertainty arise primarily from incoming orders, not from the picking process and thus these works do not update decisions due to uncertain synchronization times.

Another emerging and related area is research on store fulfillment operations (Hübner et al. 2022, Bayram and Cesaret 2021, Lin et al. 2021, Gallino and Moreno 2014, Gao and Su 2017, He et al. 2021, Jin et al. 2018, Li 2020, Yang and Zhang 2020). Past research focuses on inventory management (Chen et al. 2019), store allocation decisions (Das et al. 2023, Vazquez-Noguerol et al.

2022, Dethlefs et al. 2022), cost of service (He et al. 2020, Liu et al. 2023, Ni et al. 2019), and customer satisfaction (Wang et al. 2022). Notably, operational decisions inside the store have been largely ignored; exceptions include (MacCarthy et al. 2019, Masel and Mesa 2018, Zhang et al. 2019, Zhang and Pazour 2019, Mou 2022a, Difrancesco et al. 2021), but none use in-store customers or cobots. Only limited research has explored the intralogistic tasks of order picking in stores (Seghezzi et al. 2022, Salgado 2015, Pietri et al. 2021, Neves-Moreira and Amorim 2023, Mou 2022b), and all are manual operations without the use of automation. No previous work explores the use of cobots in order picking processes in a store environment. Related is work that investigates deploying cobots for intralogistics tasks in a store environment for inventory replenishment (Caporaso et al. 2022). Additionally, crowdsourced order picking has been explored where in-store customers pick items while doing their own shopping (Dayarian and Pazour 2022). However, this paper focuses on assignment of orders (not routing of a set of resources), nor does it consider the utilization of cobots. Therefore, this is the first paper to consider the use of cobots for store fulfillment in a retail setting and is also the first to study cobots within a crowdsourced setting. The contribution of this work is thus in modeling and developing a solution approach for resource dispatching, routing, and abandonment capturing the salient features of this unique order fulfillment environment.

3. Problem Statement

We consider a store that receives online orders spontaneously over time. Each online order consists of a set of items found on the store's shelves and is expected to be available for last-mile delivery or curbside pickup at a designated dropoff station, denoted as v_p, within a given service guarantee (e.g., in 2 hours, next day). To fulfill these online order requests, the store uses a collaborative process where in-store customers extract items from store shelves and place them in waiting cobots to transport the picked items back to the dropoff station. To ensure high service levels, the store also deploys a set of dedicated pickers to help pick and transport items from store shelves. The store has control over the dedicated pickers and the cobots, and these resources can be instructed to complete specific tasks in a specific sequence. This is in contrast to the in-store customers, who help with the picking process but the store does not have control over their actions. Thus, a store has a set K of fulfillment resources available and under their control, $K = C \cup D$, where C is the set of cobots and D is the set of dedicated pickers.

As is common in warehousing order fulfillment operations (Ardjmand et al. 2018, Liang et al. 2020, Rasmi et al. 2022), the store deploys a wave system, which splits the workday into discrete, equally spaced time periods, known as waves each of length T. The store fulfillment process is segregated into three separate work processes, each with its own dedicated resources that run in parallel. These processes include (1) receiving and dispatching online orders (including performing a check of requested items versus point-of-sale inventory levels and determining suitable substitute products in case of out-of-stock items), (2) traveling to, picking, and transporting requested items from store shelves to a dropoff station, and (3) sorting, packing, quality control, and interfacing with customers for order pickup. The focus of this paper is on optimizing the second process step. The store collects online order requests that have arrived at the store over the previous wave(s) and are made available to be assigned to the set of controllable picking resources in a future wave based on meeting service deadlines. This means at the beginning of a wave the store has a known set of items, all with the same urgency that needs to be retrieved from store shelves and returned to the dropoff station by the end of the wave (i.e., within the wavelength T).

The store fulfillment problem using cobots and dedicated pickers can be defined on a graph $\mathcal{G} = \{V, \mathcal{E}\}$, where $V = V^s \cup v_p \cup v_g$ are the sets of nodes and $\mathcal{E}$ are the set of edges of the graph. The store has a set of designated stopping points V^s where the cobots will travel to and from and wait (see Figure 1(a)). Each stopping point covers a specific shelf area around them, and thus, collectively across all stopping points in the store, all items in the store are covered by the set of stopping points, V^s. All picking resources, $k \in K$, will start their picking route from the idle station v_g, and end their wave's route at the dropoff station, v_p. The travel times along the edges of the graph depend on the picking resource type, with t^D_{ij} and t^C_{ij} being the travel time of a dedicated picker and a cobot along edge $(i, j) \in \mathcal{E}$, respectively. During a given wave, all stopping points may not need to be visited; thus, the set of stopping points required to be visited during the next wave is denoted as $V^r \subseteq V^s$. Hence, $\hat{V} = V^r \cup v_p \cup v_g$ is the subset of nodes of graph $\mathcal{G}$ that are present in the targeted wave. The number of items required to be picked from stopping point $v \in V^r$ is denoted by n_v, and collectively across all V^r the number of items required to be picked in a given wave is denoted by $N = \sum_{v \in V^r} n_v$.

When a cobot $k \in C$ reaches a stopping point $v \in V^r$, the cobot will stop and wait for a participating in-store customer's arrival which we refer to as waiting time. If a participating in-store customer arrives at the covered area of a $v \in V^r$ where a cobot is waiting, a synchronization of a cobot and in-store customer with the completion of n_v picks occurs. Thus, the wait time of a cobot $k \in C$ at a stopping point $v \in V^r$ before being synchronized with an in-store customer is a random value denoted by $\bar{\omega}_{kv}$. Simultaneously, the fleet of dedicated pickers is deployed by the store to pick a subset of items in V^r and then transport them to v_p, but they do not need any synchronization requirement to complete their picks (i.e., $\bar{\omega}_{kv} = 0$ for $k \in D$). The random service time at a given stopping point $v \in V^r$ that captures the picking time per item is denoted by $\bar{s}_v$.

The purpose of the *stochastic order-picking problem with uncertain synchronization times of in-store customers and collaborative robots* is to identify a set of routes starting at v_g and ending at v_p for the set of dedicated pickers and the set of cobots such that the total number of items picked is maximized and the picking resources are back to v_p within the wavelength T. To achieve this objective, the store adopts a centralized decision-making mechanism to determine how best to utilize its controllable picking resources during a given wave.

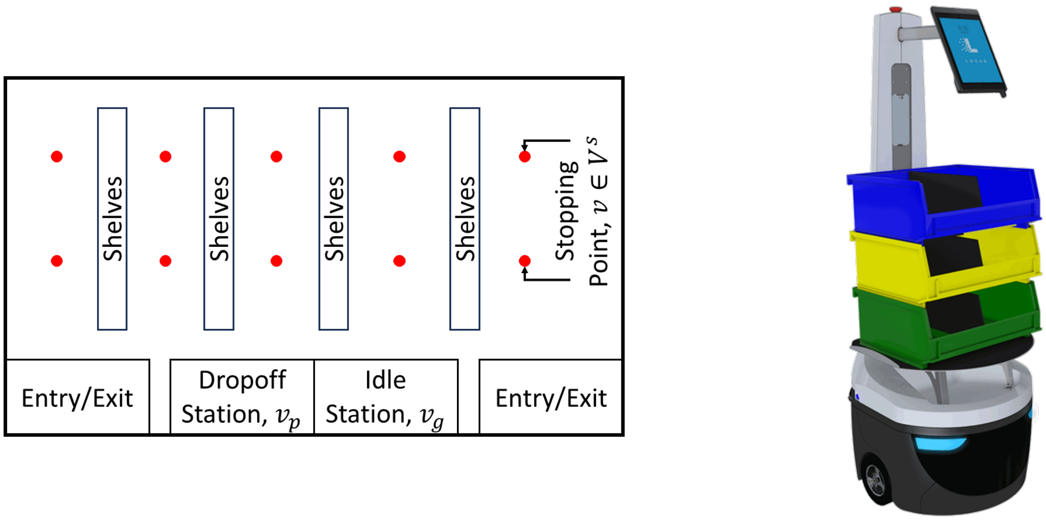

Figure 1 **From left to right (a) example store layout with stopping points (b) Commercially available AMR collaborative robot (source: Locus Robotics)**

4. Modeling as Markov Decision Process

We model our problem setting as a Markov Decision Process (MDP) with the objective to maximize the total expected number of picked items returned back to v_p within T, given the set of resources $k \in K$ and uncertain arrival times of in-store customers. We are interested in determining, over the horizon of a wave, a picking policy, which consists of a set of sequential decisions about which cobots and which dedicated pickers should be deployed to pick which items, and in what order, and when such resources should abandon their current stopping point, and when they should return back to the dropoff station. In the next subsections, we describe the elements of the MDP model.

4.1. Decision Epochs

The system makes a decision based on the updated information at every decision epoch, which occurs anytime one of the triggering events (τ^u) in (1) takes place.

$$\tau^u = \begin{cases} \tau^I & \text{the initial trigger, which occurs at the beginning of the wave;} \\ \tau^c_k & \text{when an in-store customer has dropped } n_v \text{ items into a } k \in C\text{'s tote at any } v \in V^r; \\ \tau^d_k & \text{when a dedicated picker } k \in D \text{ completes picking } n_v \text{ items at any } v \in V^r; \\ \tau^a_k & \text{whenever a cobot } k \in C \text{ needs to consider abandoning at any } v \in V^r; \\ \tau^r_k & \text{whenever any } k \in K \text{ needs to travel to } v_p \text{ to meet the wave deadline;} \\ \tau^f & \text{when all } k \in K \text{ reaches } v_p. \end{cases} \tag{1}$$

Given a finite number of these events can occur over the wavelength, we have a finite set of discrete decision epochs, which are revealed dynamically during a wave. Let $E = \{e_1, e_2, ..., e_{|E|} = \tau^f\}$ be the set of decision epochs.

4.2. Rewards

Whenever the system reaches a state $s_e \in S$ at epoch $e \in E$, an action is chosen and an immediate reward is accrued. This reward, denoted as $\beta(s_e, \alpha)$, is a function of two elements - the state of the system at epoch e and the type of action taken.

4.3. States

The state of the system $s_e \in S$ at decision epoch $e \in E$ can be described by a tuple, i.e., $s_e = \langle t_e, \left(l_{ke}\right)_{k\in K}, \left(z_{ve}\right)_{v\in V^r}\rangle$. Element t_e is the time at which the decision epoch $e \in E$ was triggered, l_{ke} is the locations of picking resources $k \in K$ in the store at epoch $e \in E$, and z_{ve} is the status of stopping point $v \in V^r$ at t_e, which we denote using a binary parameter defined as

$$z_{ve} = \begin{cases} 0 & \text{if the assigned } n_v \text{ items have all been picked at } v \in V^r; \\ 1 & \text{if a picking resource has not been assigned to pick at } v \in V^r. \end{cases} \tag{2}$$

4.4. Actions

When a new decision epoch, $e \in E$, is triggered, an action α is taken that causes the system to transition from the current state s_e to the next state s_{e+1}. There are four types of actions:

$$\alpha = \begin{cases} \alpha_1 & \text{send the triggered picking resource } k \in K \text{ to a } v \in V^r \text{ that has its } z_{ve} = 1 \text{ at any} \\ & \tau_k^c,\ \tau_k^d, \text{ or } \tau_k^a; \\ \alpha_2 & \text{send the triggered picking resource } k \in K \text{ to } v_p \text{ at any } \ \tau_k^r\ ; \\ \alpha_3 & \text{send all picking resources to their first assignments at epoch } e_1 \text{ (i.e., } \tau^I); \\ \alpha_4 & \text{keep the triggered picking resource } k \in K \text{ waiting at current } v \in V^r \text{ at any} \\ & \tau_k^a. \end{cases} \tag{3}$$

We assume that no two stopping points $v \in V^r$ would change their status from $z_{ve} = 1$ to $z_{ve} = 0$ at the same time. Additionally, we assume the set of actions occur assuming all picking resource(s) have reached their previously made decisions (we do not allow resources en route to be diverted by an action). Also, actions that assign more than one picking resource to a stopping point at the same time are prohibited.

4.5. Transition Probabilities

Depending on the state and action, the system transitions from state s_e at epoch e to another state, s_{e+1} at epoch $e+1$. We break down this transition into two separate steps: (1) a post-decision state, and (2) a pre-decision state. First, from s_e, the system transitions to a post-decision state

s_e^α reflecting (but not yet reached) three aspects: the new assignment(s) (i.e., new locations of the picking resources), time when the triggered picking resource will reach its new location, and the status of $v \in V^r$. From the post-decision state s_e^α, the system transitions to a pre-decision state s_{e+1} at epoch $e+1$ in a probabilistic manner, i.e., the probability that one of the triggering events $(\tau_k^c, \tau_k^d, \tau_k^r, \tau_k^a)$ occur before the others. Such transition probabilities have been categorized and defined in Appendix A.

4.6. Objective Function and Optimal Policy

The objective of our problem is to maximize the number of items picked and transferred to v_p within T. As there are a finite set of states S as well as a finite set of actions at each decision epoch e, an optimal deterministic Markovian policy is existent (Puterman 2014). A policy π here can be defined as a sequence of actions for each decision epoch in the wave. An optimal policy $\pi^* \in \Pi$ would therefore take the form of Equation (4) that refers to achieving the maximum expected sum of rewards given the initial state of s_{e_1} where α_e^π denotes actions following policy π at epoch e.

$$\pi^* = \arg\max_{\pi \in \Pi} \mathbb{E}\left[\sum_{e=e_1}^{e_{|E|}} \beta\left(s_e, \alpha_e^\pi\right) | s_{e_1}\right] \tag{4}$$

5. Solution Approach

MDPs are notorious for being computationally expensive for most practical problems (Ulmer et al. 2020). Thus, we develop a heuristic solution approach framework that decomposes the problem into tractable decision stages. This approach also enables us to explore the impact on picking performance of alternative methods for the different stages (see Section 6 for computational results across the resulting 12 different solution variants).

Our solution approach is designed so all cobots and dedicated pickers have an initial set and sequence of stopping points and return times, and cobots have an expected amount of time to wait before abandoning their currently assigned stopping point. Due to the high levels of uncertainty associated with cobot and in-store customer synchronization times, a key feature of the heuristic is to update these decisions as new information becomes available. Thus, in subsections that follow, we first describe the approach to initialize the set, sequence, wave-time, and abandon triggers of stopping points for each of the resources, and then how decisions are updated during the wave in the face of the materialized synchronization times.

5.1. Initialization

As shown in Figure 2, the initialization block uses a decomposition approach where we first decide on the assignment of stopping points to the set of resources. We then determine each resource's initial sequence of the assigned stopping points. Based on these sequences, we determine the wave-time triggers for all resources, and the abandon triggers for all cobots. Lastly, we deploy the resources to their first assignments.

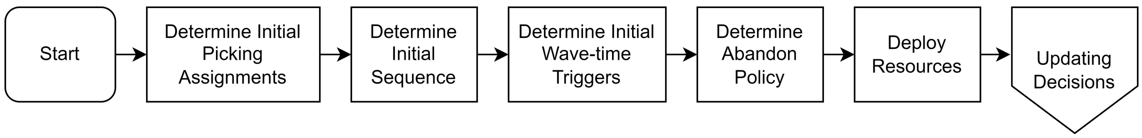

Figure 2 **Initialization block for the heuristic approach**

5.1.1. Determine Initial Picking Assignments: We take a holistic view of the wave's picking tasks and different types of resources to determine the initial assignment of picking points to resources that (i) balances the expected workload across picking resources, and (ii) captures that cobots require different expected synchronization time for different stopping points, while dedicated pickers require no synchronization time. To do so, at $\tau^I(t_e = 0)$, we find the (initial) cluster of stopping points $v \in V^r$ that each resource $k \in K$ should visit by solving the Linear Integer Program in (7)-(13). This program minimizes the maximum time expected to travel, wait, and pick all items across the stopping points $v \in V^r$ and return back to the dropoff station. As minimizing the exact traveling times within a cluster is computationally expensive (Nascimento et al. 2010), this formulation is motivated by the MIP-Diameter problem (Sağlam et al. 2006), which minimizes the maximum traveling distance within a cluster.

Inputs to our model include for each picking resource $k \in K$, their expected waiting time ω_{kv} (an approximation of $\bar{\omega}_{kv}$), their expected picking time at stopping point $v \in V^r$ denoted by $s_v n_v$ (s_v is an approximation of $\bar{s}_v$), and a surrogate travel time. The surrogate travel time, t^h_{kv}, is calculated as the average travel time spent by picking resource $k \in K$ from a $v \in V^r$ to all required stopping points and stations $\hat{V} \setminus \{v\}$ using equation (5) and (6).

$$t^h_{kv} = \frac{\sum_{j\in\hat{V}\setminus\{v\}} t^C_{vj}}{|\hat{V}|-1} \quad \forall v \in V^r, k \in C \qquad (5) \qquad\qquad t^h_{kv} = \frac{\sum_{j\in\hat{V}\setminus\{v\}} t^D_{vj}}{|\hat{V}|-1} \quad \forall v \in V^r, k \in D \qquad (6)$$

We define decision variables x_{kv} having value 1 if picking resource $k \in K$ will initially be assigned to visit $v \in V^r$ and 0 otherwise. Decision variables t^M_k express the expected amount of time by picking resource $k \in K$ for their combined waiting, picking, and surrogate traveling, and decision variable Z^Δ expresses the expected amount of time the last resource will return back to the dropoff station after completing their picking assignments.

$$\min \quad Z^\Delta \qquad (7)$$

$$\text{s.t.} \quad Z^\Delta \geq t^M_k; \qquad \forall k \in K; \qquad (8)$$

$$t^M_k = \sum_{v\in V^r} x_{kv}\omega_{kv} + \sum_{v\in V^r} x_{kv}s_v n_v + \sum_{v\in V^r} t^h_{kv}x_{kv}; \qquad \forall k \in K; \qquad (9)$$

$$\sum_{k\in K} x_{kv} = 1; \qquad \forall v \in V^r; \qquad (10)$$

$$x_{kv} \in \{0,1\}; \qquad \forall v \in V^r, k \in K; \tag{11}$$

$$t_k^M \geq 0; \qquad \forall k \in K; \tag{12}$$

$$Z^\Delta \geq 0. \tag{13}$$

The objective function in (7) minimizes the maximum expected pick completion time across all resources by enforcing in (8) that Z^Δ be greater than or equal to $t_k^M \forall k \in K$. The expected completion time of each resource is a function of which $v \in V^r$ are assigned to which resource, as enforced in (9). Constraints (10) ensure that each stopping point is assigned to one and only one picking resource. Lastly, constraints set (11) ensure that all x_{kv} hold only binary values, constraints set (12) and (13) ensures non-negative values. After solving (7) - (13), we define the initial non-sequenced assignments of stopping points for each picking resource $k \in K$, denoted by V_k^Λ, using (14).

$$V_k^\Lambda = \{v \in V^r | x_{kv} = 1\} \quad \forall k \in K \tag{14}$$

5.1.2. Determine Initial Sequences: We consider two alternatives to determine the initial sequence of the assignments in $V_k^\Lambda \quad \forall k \in K$. The first one prioritizes minimizing travel time $\forall k \in K$ and the second alternative prioritizes visiting lower synchronization time points earlier $\forall k \in C$.

Alternative 1: TSP-Based: The first alternative minimizes the travel time for each $k \in K$ by solving a separate Traveling Salesman Problem (TSP) that sequences V_k^Λ in terms of the minimum travel time of their route that starts at v_g and ends at v_p.

Alternative 2: Ranking-Based: The ranking-based alternative sequences stopping points for cobots in a descending manner of $b_v n_v$ (or equivalently ascending order of expected wait times per pick) for each $v \in V_k^\Lambda \ \forall k \in C$. Because all dedicated pickers have zero wait time, their sequences are determined using a TSP-based approach (similar to alternative 1).

For both alternatives, the output is a sequenced set of stopping points denoted by $R_k(t_e) = \{R_{k1}(t_e), R_{k2}(t_e), ...\} \forall k \in K$, where $R_{ki}(t_e)$ is the i^{th} stopping point in the sequence updated at t_e. We also define $\theta_k(t_e)$ as the current assignment for resource k updated at t_e in (15).

$$\theta_k(t_e) = R_{k1}(t_e) \quad \forall k \in K \tag{15}$$

5.1.3. Determine Initial Wave-Time Triggers: We set the wave time triggers for each picking resource $k \in K$ so that all picking resources return to the dropoff station, v_p, by T but no earlier (unless all items have been picked). To do this, we denote the wave-time trigger for picking resource $k \in K$ updated at time t_e as $\tau_k^r(t_e)$. This is set using equation (16) and (17), where $t_{\theta p}^C$ and $t_{\theta p}^D$ are the travel time required from the current assignment $\theta_k(t_e)$ to v_p for a cobot ($t_{\theta p}^C$), and a dedicated picker ($t_{\theta p}^D$), respectively.

$$\tau_k^r(t_e) = T - (t_{\theta p}^C) \quad \forall k \in C \tag{16}$$

$$\tau_k^r(t_e) = T - (t_{\theta p}^D) \quad \forall k \in D \tag{17}$$

Therefore, when the time $\tau_k^r(t_e)$ is reached at epoch $e+1$, $\theta_k(t_{e+1})$ is set to v_p; that is, the corresponding picking resource is sent to the dropoff station to ensure the resource arrives back by the wavelength.

5.1.4. Determine Abandon Policy Triggers: As a result of stochasticity in synchronization times, a cobot might end up waiting for a long period of time which reduces the time available to retrieve the remaining items in the cobot's sequence. To mitigate this risk, we create an abandonment policy, where a cobot may abandon their current point because leaving and going to the next stopping point provides a higher expected value of items picked and returned to the v_p by T. To quantify the value of such an abandon policy, we explore two solution approach variants: one with and one without an abandon policy.

Alternative 1: Using Abandon Policy

To decide on the maximum waiting time a given cobot $k \in C$ should continue waiting at their current stopping point $\theta_k(t_e)$, we develop a constrained optimization model that is solved independently for each $k \in C$. The decision variables are the allowable waiting times at stopping point $v \in R_k(t_e)$ denoted as t_v^ϵ. Input parameters are n_v, T, and t_e.

$$\max \quad \sum_{v \in R_k(t_e)} F_v n_v \tag{18}$$

$$\text{s.t.} \quad \sum_{v \in R_k(t_e)} t_v^\epsilon \leq T - t_e; \tag{19}$$

$$t_v^\epsilon \geq 0; \qquad \forall v \in R_k(t_e). \tag{20}$$

The objective function in (18) maximizes the expected number of items picked and returned given the remaining time in the wave. Here, F_v denotes a function to calculate the probability of at least one participating customer showing up within t_v^ϵ amount of time. In (19), the total waiting allowed for all stopping points remaining in the resource's sequence is the time remaining in the wave. Lastly, constraints (20) enforce a non-negative waiting time at each $v \in R_k(t_e)$. Thus, at τ^I, we solve (18)-(20) for all $k \in C$ and set the abandon trigger following equation (21).

$$\tau_k^a(t_e) = t_e + t_{\theta_k(t_e)}^\epsilon \quad \forall k \in C \tag{21}$$

Alternative 2: No Abandon Policy: When the framework does not use an abandon policy, we simply set the initial abandon triggers to the wavelength (i.e., $\tau_k^a(t_e) = T \quad \forall k \in C$), and do not update them during the wave.

5.1.5. Deploy Resources: The last step in initialization is to send all resources $k \in K$ to their first assigned point, $\theta_k(\tau^I)$. This is action α_3 which occurs at epoch e_1. Then the solution approach waits for a new triggering event.

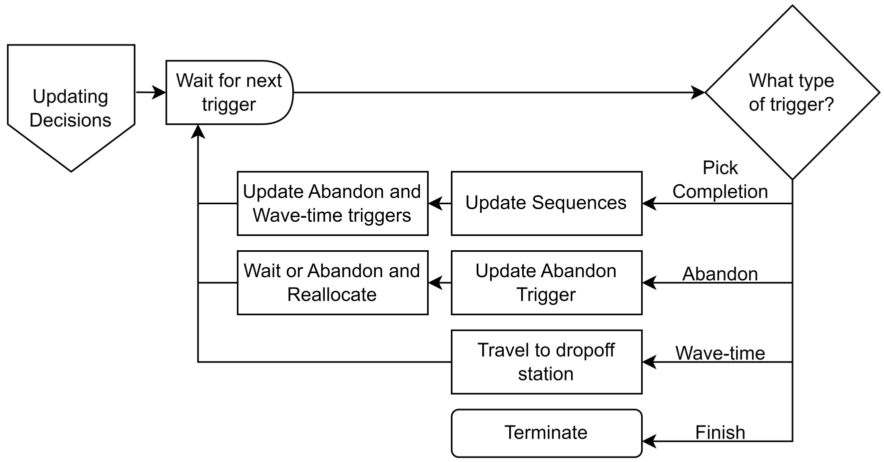

Figure 3 **Updating decisions block for the heuristic approach**

5.2. Updating Decisions

Whenever any triggering event occurs, we update decisions depending on which triggering event was observed (see Figure 3). We denote the triggered resource at time t_e as $k^{\psi}(t_e)$.

5.2.1. Triggered by a Pick Completion (τ_k^c or τ_k^d) :

After a pick completion trigger, the approach considers whether or not to move not-yet visited picking locations among the set of resources. If the system reaches a pick completion trigger (i.e., $\tau^u = \tau_k^c$ or τ_k^d for any $k \in K$) at epoch $e \in E$, we first remove the picked stopping point from the sequence of k^{ψ} and leave the other resources' sequences as is. This results in the pre-decision sequences $R_k^q(t_e)$ given in equation (22).

$$R_k^q(t_e) = \begin{cases} R_k(t_{e-1}) \setminus \{\theta_k(t_{e-1})\} & \text{if } k = k^{\psi} \\ R_k(t_{e-1}) & \text{if } k \neq k^{\psi} \end{cases} \tag{22}$$

Next, we follow the below-mentioned steps to transition from pre-decision sequences, $R_k^q(t_e)$ to post-decision sequence, $R_k(t_e)$.

Step 1: Using equation (23) and (24), we calculate the expected pick completion time, t_k^{σ}, for each $k \in K$, which is the sum of the travel time, the expected waiting time, and the expected picking time at each $v \in R_k^q(t_e)$, added to the epoch's trigger time t_e. Let, $R_{kN}^q(t_e)$ denote the last stopping point to visit in the sequence of $k \in K$. Also, we calculate t_k^Q, the slack time for each $k \in K$, using equation (25).

$$t_k^{\sigma} = \sum_{i=R_{k1}^q(t_e)}^{R_{k,N-1}^q(t_e)} \sum_{j=R_{k2}^q(t_e)}^{R_{kN}^q(t_e)} t_{ij}^C + \sum_{v \in R_k^q(t_e)} \omega_{kv} + \sum_{v \in R_k^q(t_e)} s_v n_v + t_e \;\; \forall k \in C \tag{23}$$

$$t_k^{\sigma} = \sum_{i=R_{k1}^q(t_e)}^{R_{k,N-1}^q(t_e)} \sum_{j=R_{k2}^q(t_e)}^{R_{kN}^q(t_e)} t_{ij}^D + \sum_{v \in R_k^q(t_e)} \omega_{kv} + \sum_{v \in R_k^q(t_e)} s_v n_v + t_e \;\; \forall k \in D \tag{24}$$

$$t_k^Q = \max_{k\in K}(t_k^\sigma) \; - t_k^\sigma \quad \forall k \in K \tag{25}$$

Step 2: Identify two disjoint sets of resources: (i) resources to consider adding stopping points (denoted as the set K^H) and (ii) resources to consider removing stopping points (denoted as the set K^m). Any resource not in K^m are resources with the potential for add stopping point(s), which we define as $K^H = K \setminus K^m$. The associated stopping points in the sequences of K^m is denoted by V^m and defined by $V^m = \bigcup_{k\in K^m} R_k^q(t_e)$ and the stopping points of K^H is denoted by V^H and defined by $V^H = \bigcup_{k\in K^H} R_k^q(t_e)$

$$K^m = \begin{cases} \arg\max_{k\in K}(t_k^\sigma) & \text{if } |\{k \in K : t_k^\sigma > T\}| = 0 \\ \{k \in K | t_k^\sigma > T\} & \text{if } |\{k \in K : t_k^\sigma > T\}| \geq 1 \end{cases} \tag{26}$$

Step 3: Define set G which represents the set of detour positions within each sequence of a resource $k \in K^H$. For instance, for any $k \in K^H$, the first detour position will be in between $R_{k1}^q(t_e)$ and $R_{k2}^q(t_e)$.

Step 4: Solve the multi-objective Mixed-Integer Linear Programming (MILP) model in (27) to (37) to determine the decision variable values: (i) x_{ij} which is 1 if stopping point $i \in V^m$ is inserted into position $j \in G$ and 0 otherwise; (ii) t_k^ϕ which denotes the updated expected pick completion time for picking resource $k \in K$; and (iii) Z^Γ which denotes the maximum value among all t_k^ϕ. Input parameters are B_{jk}, which is 1 if stopping point $j \in G$ is in the sequence of $k \in K^H$ and 0 otherwise, t_{ij}^+ which expresses the amount of additional expected time to account for adding stopping point $i \in V^m$ in the position $j \in G$, (i.e., the expected detour time) which includes additional traveling, waiting and picking time after insertion, and t_{ik}^- which denotes the amount of expected pick completion time to be deducted from the sequence of picking resource $k \in K^m$ if stopping point $i \in V^m$ is moved to another resource's sequence.

$$\min \quad Z^\Gamma \tag{27}$$

$$\min \quad \sum_{k\in K} t_k^\phi \tag{28}$$

$$\text{s.t.} \quad \sum_{i\in V^m}\sum_{j\in G} x_{ij} t_{ij}^+ B_{jk} \leq t_k^Q; \qquad \forall k \in K^H; \tag{29}$$

$$\sum_{i\in V^m} x_{ij} \leq 1; \qquad \forall j \in G; \tag{30}$$

$$\sum_{j\in G} x_{ij} \leq 1; \qquad \forall i \in V^m; \tag{31}$$

$$\sum_{i\in V^m}\sum_{j\in G} x_{ij} \leq 1;; \tag{32}$$

$$t_k^\phi = t_k^\sigma + \sum_{i\in V^m}\sum_{j\in G} x_{ij} t_{ij}^+ B_{jk}; \quad \forall k \in K^H; \tag{33}$$

$$t_k^{\phi} = t_k^{\sigma} - \sum_{i \in V^m} \sum_{j \in G} x_{ij} t_{ik}^{-}; \qquad \forall k \in K^m; \tag{34}$$

$$t_k^{\phi} \leq Z^{\Gamma}; \qquad \forall k \in K; \tag{35}$$

$$t_k^{\phi} \geq 0;; \tag{36}$$

$$Z^{\Gamma} \geq 0. \tag{37}$$

The primary objective function (27) minimizes Z^{Γ}, which in conjunction with constraint set (35) minimizes the maximum t_k^{ϕ}. When there are multiple optimal solutions all achieving the same primary objective function value, then we prioritize solutions using the secondary objective function (28), which minimizes the sum of all t_k^{ϕ}. Constraint set (29) ensures that the expected pick completion time addition to picking resource $k \in K^H$ does not exceed the limit of t_k^Q. Constraint set (30) ensures that each detour position is filled by at most one stopping point from V^m. Similarly, constraints set (31) ensures that each stopping point in V^m is reallocated into at most one detour position. The set of constraints (32) limits the amount of total reallocations performed to only one and this constraint is our source of creating alternative methods. The updated expected pick completion time is calculated in constraint set (33) for the resources in K^H and (34) for resources in K^m. Lastly, inequalities (36) and (37) ensure non-negative values of decision variables. In our solution framework, we consider three different alternatives:

- **Alternative 1: S: Single Reallocation:** We use constraints (32) to limit the total number of reallocations to one.
- **Alternative 2: M: Multiple Insertion:** We do not use constraints (32) to allow for more than one reallocations.
- **Alternative 3: Z: Zero Updates:** No reallocations are made.

After solving, the MILP (27) - (37), we get the optimal values of x_{ij}. For each values of $x_{ij} = 1$, we perform two operations: (i) we insert $i \in V^m$ to the position of $j \in G$ and (ii) we remove $i \in V^m$ from the corresponding sequence of $k \in K^m$ to reconstruct the pre-decision sequences, $R_k^q(t_e)\ \forall k \in K$, and achieve post-decision sequences $R_k(t_e)\ \forall k \in K$. Lastly, we update the abandon triggers $\tau_k^a(t_e) \forall k \in C$ where $R_k(t_e) \neq R_k^q(t_e)$ by solving the constrained optimization model in (18) - (20) and then set $\tau_k^a(t_e)$ following equation (21). Additionally, the wave-time triggers $\tau_k^r(t_e)$ for $k^{\psi}(t_e)$ are also updated following equation (16) or (17).

5.2.2. Triggered by wave-time trigger τ_k^r: Whenever the picking system reaches a wave-time trigger $\tau^u = \tau_k^r$ for any picking resource $k \in K$, $k^{\psi}(t_e)$ is instructed to return to v_p, hence $\theta_k(t_e) = v_p$ for $k^{\psi}(t_e)$. Here, $k^{\psi}(t_e)$ not only has to travel to v_p if it was waiting there for synchronization but also if it was in the process of picking item(s). Therefore, a stopping point $v \in R_k(t_e)$ may have a fraction of n_v picked by a resource.

5.2.3. Triggered by abandon trigger $\tau^a_k(t_e)$: Whenever $\tau^u = \tau^a_k$ for any $k \in C$, we update the amount of time we should wait at $\theta_k(t_{e-1})$ for $k^\psi(t_e)$ based on the abandon policy in Section 5.1.4 and update τ^a_k following equation (38). Additionally, $R_k(t_e)$ for $k^\psi(t_e)$ also gets updated using equation (39). This is then followed by updating the current assignment $\theta_k(t_e)$ of resource $k^\psi(t_e)$ using equation (15) and updating the wave-time trigger using equation (16) or (17). If the cobot is instructed to stop waiting at $R_{k1}(t_{e-1})$ (i.e., $t^\epsilon_v = 0$), the abandoned stopping point, denoted by $v^a(t_e)$ gets added to the end of any picking resource's sequence providing the least increase of maximum expected pick completion time, following equation (40). And we do not recalculate the abandon trigger after adding this point (as this stopping point had already been given the opportunity to synchronize).

$$\tau^a_k(t_e) = \begin{cases} t_e + t^\epsilon_v | v = R_{k1}(t_{e-1}) & \text{if } t^\epsilon_v > 0 | v = R_{k1}(t_{e-1}) \\ t_e + t^\epsilon_v | v = R_{k2}(t_{e-1}) & \text{if } t^\epsilon_v = 0 | v = R_{k1}(t_{e-1}) \end{cases} \tag{38}$$

$$R_k(t_e) = \begin{cases} R_k(t_{e-1}) & \text{if } t^\epsilon_v > 0 | v = R_{k1}(t_{e-1}) \\ R_k(t_{e-1}) \setminus R_{k1}(t_{e-1}) & \text{if } t^\epsilon_v = 0 | v = R_{k1}(t_{e-1}) \end{cases} \tag{39}$$

$$R_k(t_e) = R_k(t_{e-1}) \cup v^a : \min\big(\max(t^\sigma_k \forall k \in K) \text{ at } t_e - \max(t^\sigma_k \forall k \in K) \text{ at } t_{e-1}\big) \tag{40}$$

5.2.4. Triggered by pick finish trigger τ^f: This trigger is only set off when all picking resources are assigned to travel to the dropoff station, whether by completing all the assigned picks for the wave, N, or being sent to v_p as a result of one or more τ^r_k. For any of these two scenarios, when the algorithm finds $\theta_k(t_e) = v_p \quad \forall k \in K$, we terminate the algorithm.

6. Computational Experiments

We design our computational experiments to answer the following questions a store might have prior to deploying a cobot fulfillment strategy: (1) How much advantage is there to deploying dynamic vs static assignment policies? (2) In which situations is there value in a store deploying a cobot abandon policy? (3) How does the allowed number of reallocation of stopping points impact picking performance? (4) How are picking tasks distributed among the dedicated pickers and the cobots? (5) How does the participation rate of in-store customers affect such policies? (6) Are our solution approaches computationally tractable for decision-making in practical settings? (7) Which policy should a store utilize? (8) How many cobots are needed to replace one dedicated picker? And lastly, (9) Is the proposed approach economically viable?

To answer these questions, we evaluate the performance of our policy variants on a set of computational experiments that integrate actual online order data, store layouts and empirical consumer shopping behavior data. Our primary performance indicator is calculated by $\rho = \frac{n^P}{N}$ representing the percentage of items released at the beginning of the wave that are picked and returned back to the v_p by the end of the wave. Here, n^P denotes the number of total items picked and returned to v_p within T across all resources.

We use a three-letter acronym to denote each of the policy variants, with the first letter referring to the initial sequencing alternative (R - Ranking, T - TSP) (Section 5.1.2), the second letter for abandon policy (A - Abandon policy, N - No abandon policy) (Section 5.1.4), and the third letter for the updating alternative (S - Single Insertion, M - Multiple Insertion, Z - Zero updates) (Section 5.2.1). This leads to 12 different policy variants (See Table 1 for a summary of these variants and their features). Notably, the variants ending with NZ are static variants that make initial decisions at the beginning of the wave and do not update these decisions throughout the wave.

The computational experiments use the grocery store layout described in Hui et al. (2009), and empirical online order data from Instacart (2017). We map the product categories in the Instacart data set to the zones in Hui et al. (2009)'s store layout, and assume each of the 134 product categories corresponds to a stopping point (i.e., $|V^s| = 134$) (see Appendeix C). We set 0.6 m/s (Lee and Murray 2019) as the traveling speed of the dedicated human pickers traveling with a picking cart. For the cobots, in a warehouse environment usually 0.9-1.0 m/s is considered safe (Barcoding Inc. 2019); however, as our cobots will be operating in a retail environment, we set 0.4 m/s as the average cobot traveling speed. Combining these speeds with the layout and distances in Hui et al. (2009), we obtain t^C_{ij} and t^D_{ij}. Furthermore, we consider the mean per item picking time to be the same for all product categories and set $s_v = 25$ seconds $\forall v \in V^s, \forall k \in K$, which captures both the searching and picking time once a dedicated or in-store customer is at the stopping point (Zhang et al. 2023, Tompkins et al. 2010). Finally, we set the wavelength to be 30 minutes, i.e., $T = 1800$ seconds.

We generate 10 different order profile instances (see Appendix D), using actual online order data from Instacart (2017). These instances represent multiple different orders that have been assigned to be picked in a wave. The instances have varying levels of items to be picked, and we arranged them in ascending order of N (i.e., instance 1 has the lowest N and instance 10 the highest). In addition to differences in variations in the number of items to pick N, each instance differs in the set of stopping points to be visited (V^r) and the number of items to be picked from each stopping point (n_v).

The mean arrival rates of in-store customers willing to help with fulfillment at stopping point $v \in V^s$, denoted by λ_v is given in (41). This assumes in-store customers arrive at the store and then visit subsequent stopping points for their own shopping following a Poisson process. Here, n^ζ is the average number of customer arrivals to the store, b_v is the average purchase rate by in-store customers at stopping point $v \in V^r$, and f denotes the estimated average participation rate from in-store customers in helping the cobots. We use (42) to estimate the expected waiting time for a cobot at a stopping point, ω_{kv}, which also relies on input d_v, the average dwell time of in-store customers at the area covered by $v \in V^r$. Here, n^ζ is estimated to be about 192 customers per hour on average for Walmart stores (Thomas Ozbun 2023, Walmart 2023b, Kelly Tyko 2021), and b_v and d_v are estimated from empirical data in Hui et al. (2009).

$$\lambda_v = \frac{n^\zeta f b_v}{T} \quad \forall v \in V^r \tag{41}$$

$$\omega_{kv} = \frac{1}{\lambda_v} - d_v \quad \forall v \in V^r, k \in K \tag{42}$$

Thus, we transform F_v in (18) in the abandon policy as $F_v = 1 - e^{\lambda_v t_v^\epsilon}$ which is the Poisson Cumulative Distribution Function for at least one participating in-store customer showing up. This makes (18)-(20) a non-linear programming model, which we solve using the Sequential Least Squares Programming (SLSQP) method (Kraft 1988). This is a gradient-based method that has been shown to produce high quality solutions quickly (Sahin 2019). This non-linear optimization approach requires an initial solution, and we set the initial solution using equation (43).

$$t_v^\eta = \Big(\frac{b_v n_v}{\sum_{v \in R_k(t_e)} b_v n_v}\Big)(T - t_e) \quad \forall v \in R_k(t_e) \tag{43}$$

We consider different rates of participating customers, specifically, considering 100, 125, 150, and 175 expected participating customers out of the 192 mean arrival of customers to the store. This results in 4 levels of $f = 52\%, 65\%, 78\%$, and 91%. For each level of f, we generate 100 random instances of in-store customer arrival to each $v \in V^s$.

We solve TSPs for cobots (Section 5.1.2) following a column generation method where at each iteration a master problem as well as a subproblem is solved optimally following a bidirectional labeling algorithm with dynamic halfway points (Tilk et al. 2017) which had quick convergence. As dedicated pickers get assigned more $v \in V^r$ compared to cobots as a result of no waiting time, following the same method had slow convergence. Thus, we followed a compound method where we take the best result out of an algorithm described by Santini et al. (2018) and Clarke and Wright saving algorithm (Clarke and Wright 1964) to determine the initial sequence for dedicated pickers. Lastly, we ran our experiments on a computer with processor Intel(R) Core(TM) i7-10510U CPU @1.80GHz to 2.30 GHz, 32.0 Gb of memory, Windows 11 Pro 64-bit, solver version Gurobi 10.0.1, and Python 3.11.5.

6.1. Benchmarks

For comparison purposes, we provide the pick performance achieved when a store does not deploy any cobots, but instead has 1, 2, 3, or 4 dedication pickers (denoted as a resource mix C0D1, C0D2, C0D3, and C0D4, respectively). To determine the results with only dedicated pickers, we determine the initial picking assignments as described in Section 5.1.1 and then sequence them using the TSP-based initial sequencing policy in Section 5.1.2. These decisions do not require updating because, with dedicated pickers, the work is controllable and does not change over the wave. As shown in Table 2, 1 dedicated picker (C0D1) is unable to achieve $\rho = 100\%$ in any of the instances. As we increase the number of dedicated pickers we can achieve $\rho = 100\%$ for the lower N instances only. To achieve $\rho = 100\%$ across all instances requires 4 dedicated pickers. To deploy the same total number of resources and to answer our first seven questions, in the next subsections we utilize a resource mix with 3 cobots and 1 dedicated picker (i.e., a C3D1 mix). And in Table 2 we also display the average percent picked (i.e., ρ) across each of the 10 instances, and for each of 12 variants.

		Variants											
Decision Stage	Alternatives	RAS	RAM	RAZ	RNS	RNM	RNZ	TAS	TAM	TAZ	TNS	TNM	TNZ
Initial Sequencing	Ranking(R)	x	x	x	x	x	x						
	TSP(T)							x	x	x	x	x	x
Abandon Policy	Yes(A)	x	x	x				x	x	x			
	No(N)				x	x	x				x	x	x
Updating Sequences with reallocation(s)	Single(S)	x			x			x			x		
	Multiple(M)		x			x			x			x	
	Zero(Z)			x			x			x			x

Table 1 **Decision alternatives and resulting variants**

K / Instances	C0D1	C0D2	C0D3	C0D4	C3D1											
					RAM	RAS	RAZ	RNM	RNS	RNZ	TAM	TAS	TAZ	TNM	TNS	TNZ
1	75.00%	100.00%	100.00%	100.00%	99.27%	99.33%	93.22%	99.23%	99.30%	93.22%	95.52%	95.48%	90.60%	94.98%	94.89%	90.05%
2	61.11%	100.00%	100.00%	100.00%	99.58%	99.65%	95.42%	99.56%	99.65%	95.42%	96.30%	96.47%	93.52%	96.29%	96.47%	93.57%
3	62.96%	95.06%	100.00%	100.00%	96.77%	98.21%	91.11%	96.77%	98.21%	91.11%	93.72%	93.65%	87.81%	93.60%	94.02%	87.77%
4	64.71%	95.29%	100.00%	100.00%	94.78%	96.55%	87.75%	94.59%	96.48%	87.75%	88.30%	89.55%	77.26%	87.64%	88.52%	75.44%
5	49.44%	94.38%	100.00%	100.00%	95.15%	96.88%	86.84%	95.13%	96.85%	86.84%	94.76%	93.80%	81.35%	94.66%	96.12%	80.10%
6	60.00%	92.00%	100.00%	100.00%	92.73%	94.54%	86.33%	92.69%	94.50%	86.33%	92.41%	86.00%	79.87%	92.15%	90.21%	78.90%
7	44.34%	84.91%	100.00%	100.00%	84.36%	85.77%	78.08%	84.28%	85.74%	78.08%	81.03%	81.50%	70.97%	80.87%	80.75%	70.00%
8	54.70%	85.47%	100.00%	100.00%	91.93%	93.48%	84.43%	91.90%	93.45%	84.43%	88.79%	90.61%	75.44%	88.67%	90.41%	74.57%
9	44.00%	74.40%	96.80%	100.00%	81.39%	81.37%	81.52%	81.08%	81.16%	81.52%	78.48%	76.60%	73.34%	77.75%	76.51%	71.34%
10	41.35%	74.44%	95.49%	100.00%	78.70%	81.81%	77.20%	78.58%	81.72%	77.20%	73.79%	71.58%	71.33%	73.32%	72.75%	70.14%
Overall	55.76%	89.60%	99.64%	100.00%	91.46%	92.76%	86.19%	91.38%	92.71%	86.19%	88.31%	87.52%	80.15%	87.99%	88.07%	79.19%

Table 2 **Average percent picked (ρ) for dedicated only benchmark policies and for the 12 solution variants with 3 cobots and 1 dedicated picker.**

6.2. Effect of a Dynamic vs. a Static Policy

First, we explore the impact of a store dynamically updating its resource allocation decisions within a wave. To do so, in Figure 4(a), we display the difference in the performance of a dynamic policy (i.e., the variants that end in either S or M which make updates after a pick completion trigger τ_k^c or τ_k^d) to their static counterpart that make and fix decisions at the beginning of the wave (i.e., the variants that end in Z). A positive difference means the dynamic reallocation policy performs better than its static counterpart. In Figure 4(a), all differences are positive, which means that a dynamic update of resource allocation is always beneficial. On average, a dynamic approach can provide improvements of 5.10% (across all instances) and is more valuable when a TSP-based initial sequencing policy is used versus a ranking-based initial sequencing policy. In addition to variant influences, the characteristics of the instances also influence the improvement values, and we find that improvement across all instances can range from 0.25% to 16.02%.

INSIGHT 1: Picking performance rates can be increased on average by 5.10% by dynamically updating decisions at each pick completion triggering event, rather than only making assignment decisions at the beginning of each wave, and the improvements are more pronounced for policies that use a TSP-based initial sequencing policy rather than a ranking-based one.

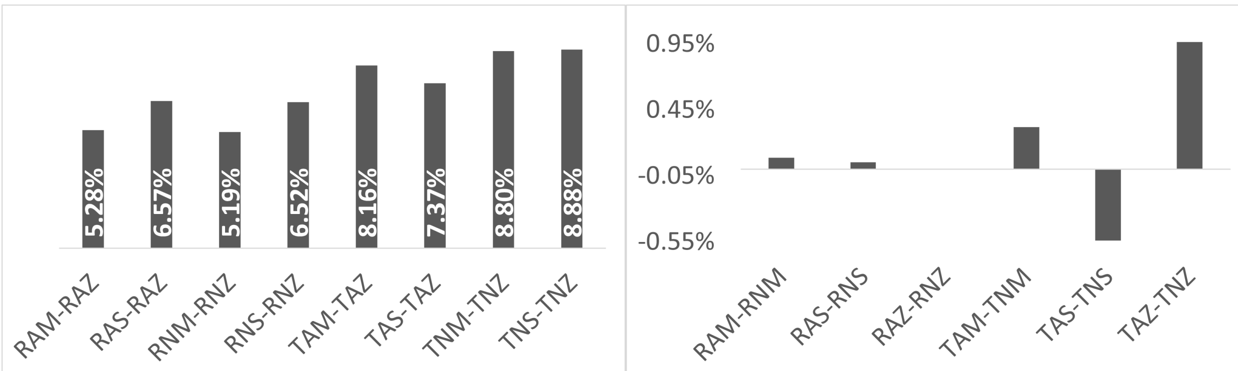

Figure 4 **From left to right, change in ρ of (a) dynamically updating resource allocations, and (b) using a cobot abandon policy.**

6.3. Effect of Cobot Abandon Policy

In Figure 4(b) we plot the improvement of performance from policies with an abandonment policy (second letter A) minus those without an abandonment policy (second letter N). For the R-variants, the decision to abandon or not has little impact on ρ. However, for the T-variants the impact is higher and the highest positive impact is achieved for TAZ-TNZ (with around a 0.95% average improvement). Such performance is significantly more efficient if the abandon policy is coupled with a dynamic reallocation after pick completion trigger policy (i.e., S or M) as TAM and TAS both have higher ρ than TAZ in Table 2. The number of abandon triggers as well as abandonment events are low in general, but more abandonment events occur for T-variants (TAZ-3578, TAM-884, TAS - 762) than R-variants (RAM - 775, RAS - 420, RAZ - 0). The numbers are for total abandonment events across all instances, which represents 4,000 waves, and thus, even the variant with the highest number of abandonment events had fewer than one event per wave.

INSIGHT 2: A cobot abandonment policy impacts the picking performance if the store deploys resources in a way that prioritizes minimizing travel distances (e.g., for policies using TSP for initial sequencing) and when such abandonment policies are combined with updating resource allocation after pick completion events. Yet, if a store deploys resources that prioritize cobot synchronization times, an abandonment policy is so seldom needed that its implementation is likely not practically warranted.

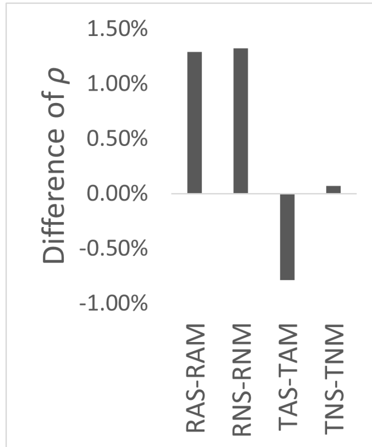

Figure 5 **Effect of reallocation number at pick completion trigger for C3D1**

6.4. Effect of Number of Reallocations at Pick Completion Trigger

Next, we explore the impact on pick performance of allowing single versus multiple reallocations of stopping points among picking resources after a pick completion trigger, i.e., after a τ_k^c or τ_k^d trigger. This is captured by the third letter in our variants, where a single insertion (S) allows at most one stopping point to be moved from a picking resource to another picking resource (see (32)), whereas (M) allows unlimited reallocations (as (32) is removed from the optimization model). To capture the effect of these contrasting policies we plot in Figure 5 the average ρ of the single(S) variant minus its multiple(M) counterpart variant; thus, positive values mean the S-variant performed better than its M-variant counterpart. S-variants perform better when coupled with ranking-based initial sequencing. Yet, multiple insertions are preferred when coupled with a TSP-based initial sequencing and an abandon policy (TAS-TAM). In the case of no abandonment policy and a TSP-based initial sequencing (TNS-TNM), multiple and single reallocations perform similarly.

INSIGHT 3: Exchanging only a single picking point among resources after a pick completion trigger provides performance benefits compared to multiple reallocation alternatives when coupled with ranking-based initial sequencing. On the contrary, allowing multiple picking points to be exchanged is beneficial after a pick completion trigger if coupled with a TSP-based initial sequencing approach and a cobot abandon policy.

6.5. Workload Distribution Among Resources

Next, to better understand how cobots are utilized, we explore how work is distributed among the dedicated picker and the cobots. To do so, in Figure 6, we plot the cobot's picking utilization, which is the ratio of items picked by all $k \in C$ and the total items picked in a wave by all $k \in K$, and the cobot's visiting utilization, which is the ratio of $v \in V^r$ visited by all $k \in C$ and all $v \in V^r$ visited by all $k \in K$. The complement of these values is the utilization for the single dedicated resource.

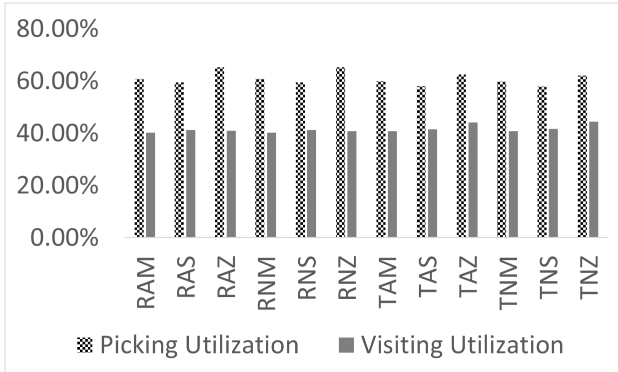

Figure 6 Utilization of cobots across all variants for C3D1 mix

f	RAM	RAS	RAZ	RNM	RNS	RNZ	TAM	TAS	TAZ	TNM	TNS	TNZ
52%	88.89%	90.00%	85.31%	88.84%	89.95%	85.31%	83.21%	82.45%	72.55%	82.90%	82.01%	71.75%
65%	90.52%	92.04%	86.19%	90.50%	91.99%	86.19%	86.69%	85.95%	79.66%	86.36%	86.77%	78.95%
78%	92.69%	94.32%	87.12%	92.55%	94.28%	87.12%	92.19%	91.32%	85.43%	92.01%	92.32%	84.66%
91%	93.76%	94.67%	86.14%	93.63%	94.59%	86.14%	91.15%	90.37%	82.97%	90.69%	91.17%	81.39%

Table 3 Effect of increasing participation rate across variants

Across the variants, the three cobots were used to pick around 60% of all items by visiting around 40% of the stopping points. As our variants prioritize cobots being assigned to stopping points with high number of picks and low expected waiting time, across all variants, cobot's picking utilization is higher than it's visiting utilization. This leads to our next insight:

INSIGHT 4: Stores can expect a significant amount of the picking tasks to be accomplished by cobots synchronizing with in-store customers: on average, across our experiments, three cobots picked around 60% of the items, resulting in the dedicated picker's workload to be the remaining 40%.

6.6. Effect of Participation Rate of In-Store Customers

In this section, we explore the impact of participating in-store customers on the variants' achieved picking performance. Table 3 breaks down ρ by participation rate. As the rate of participating customers increases, the picking rate also increases. And for the R-variants with dynamic reallocation policies after pick completion trigger (RAS, RAM, RNS, RNM), we observe an increase of picking performance as f increases. However, for the other 8 variants when $f = 91\%$ the ρ values decrease slightly compared to $f = 78\%$. For variants prioritizing minimizing traveling times or ones that do not make dynamic updates (i.e., reallocations), with such high participation rates, their decisions are made expecting a low average cobot synchronization time, but due to the uncertainty in the system, this, in many instances, is too optimistic, resulting in lower average picking performance. However, when R-variants are coupled with dynamic reallocation of stopping points, this risk is avoided and so increased expected participation rates, does not hurt ρ values.

INSIGHT 5: Ranking-based initial sequencing policies with dynamic reallocation methods are recommended for high in-store customer participation rates.

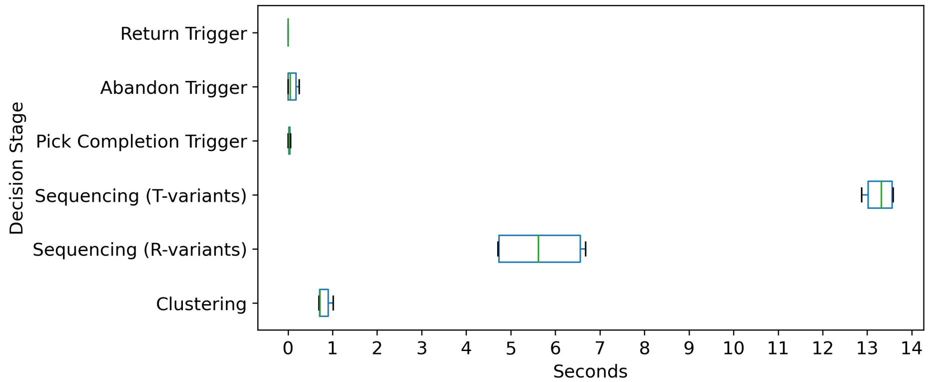

Figure 7 **Boxplots of solving time in seconds for C3D1 instances across all 12 variants, broken down by decision stage.**

6.7. Computational Time

We capture the computational time in seconds for five decision stages - clustering, sequencing, pick completion trigger, abandon trigger, and wave-time trigger (see Figure 7). Except for sequencing, the other four stages gets solved under a second, regardless of variant type. The sequencing stage requires the highest computational effort and there are large differences in computational time based on which variant is used. Even with sequential computations for each resource, the computational time for sequencing is less than 7 seconds for the R-variants, and less than 14 seconds for the T-variants. If further reductions are needed, parallel processing across resources can further reduce computational times.

INSIGHT 6: The proposed solution approaches are computationally tractable for deployment in practical settings.

6.8. Recommended Policy Variants

RAS achieves the highest average pick rate and is the best variant in 9 out of 10 instances (see Table 2). This is a policy that creates initial cobot sequences based on ranking stopping points in descending order of expected per-pick waiting times, uses an abandonment policy, and reallocates the best single stopping point dynamically. The R-variants consistently achieve improved ρ values compared to their counterpart T-variants, with average improvements ranging from 0.32% to 12.31% across the instances. As previously discussed, the store can expect a higher pick rate with this policy when there are higher rates of customers willing to participate or more customers in the store.

INSIGHT 7: Ranking-based initial sequencing policies of cobot assignments where waiting time per pick is arranged in an ascending manner results in higher pick rates than ones that create initial sequences based on minimizing total travel time.

Thus, in the remaining computational experiments, we utilize the best-performing variant, RAS.

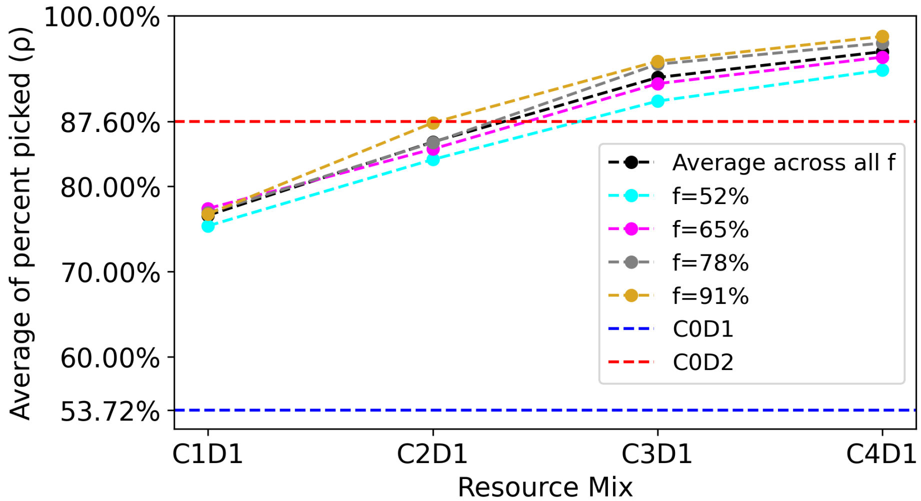

Figure 8 **Pick performance comparison with and without cobots**

6.9. Reduction in Labor Requirements with Cobots

While a store should not expect to achieve the same level of performance by replacing a dedicated picker with a cobot (due to slower travel times and stochastic synchronization times), stores can expect high performance in systems where multiple cobots augment the work of a single dedicated picker (see Table 2). Our next set of experiments are interested in understanding how many cobots are needed to replace one dedicated picker. To do so, we are interested in determining how many cobots are needed to augment the work of a single dedicated worker to achieve the same level of ρ as a system with 2 dedicated pickers but no cobots (i.e., with C0D2). Thus, we run additional experiments deploying the RAS variant with the previously defined 10 order profile instances, considering the 4 f levels, and for 100 different customer arrival patterns. Figure 8 plots ρ for different cobot levels and different in-store customer rates f. As we increase the number of cobots to assist with the picking task of one dedicated picker, the ρ value increases. A system with 2 dedicated pickers (and no cobots) achieved $\rho = 87.45\%$. This performance can be matched with one less dedicated picker and 2 cobots if $f = 91\%$ or with 3 cobots if participating rates are lower. A similar observation about C3D1 achieving equivalent performance can also be made by looking at ρ values across 10 instances (see Appendix B)

INSIGHT 8: Three cobots and a single dedicated picker can achieve equivalent picking performance as two dedicated pickers.

6.10. Economic Analysis

Lastly, we explore the economic benefit of deploying C3D1 resource mix instead of C0D2 resource mix and identify after what time period C3D1 is more beneficial than C0D2. While robotic solutions tend to have high initial costs (Marc Wulfraat 2023), there are now subscription services for cobots,

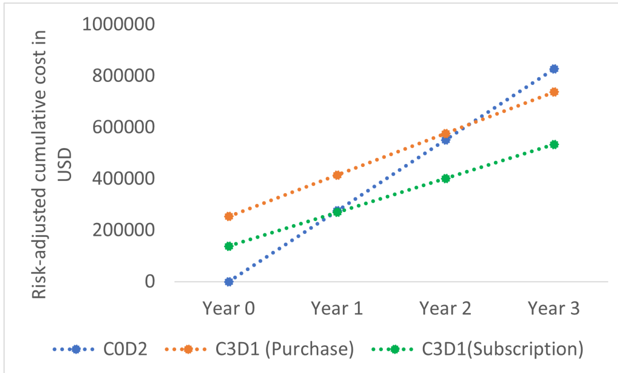

Figure 9 **Economic analysis of deploying only dedicated pickers versus investing in cobots via either direct purchase or a subscription model.**

where the store would not have to pay large upfront investment costs (Forrester Research 2019). We consider both investment scenarios: (i) a store purchases the cobots upfront and (ii) a store utilizes a subscription model. For the first option, we consider a per cobot purchasing cost of $35,000(Marc Wulfraat 2023), and annual maintenance of 20% of purchase cost (Lucas Systems 2020). For the second option, we consider a monthly subscription of $950 per cobot (Forrester Research 2019). For both options, we also consider a one time deployment fee of $75,000 and a one time integration costs of $50,000, as well as 2 hours of required training per year for each of the 12 cross-trained employees (average small team size) responsible for interacting with the cobot (Drew Holler 2021), and risk adjusted discount of 10% (Forrester Research 2019). For the dedicated pickers, we consider a hourly wage of $17.50 (Walmart 2023a), a legally required benefit per hour per employee of $2.80 (Bureau of Labor Statistics 2023), and assume the store is open 17 hours per day and 363 days per year (Kelly Tyko 2021).

Figure 9 plots the risk-adjusted cumulative costs in USD for three cases: C0D2 (the status quo of store fulfillment), C3D1 with a subscription model and C3D1 with a purchase model. Initially (beginning of year 1, referred to as year 0), the purchase model cost is higher than the subscription model and the benchmark status quo case without cobots. Yet, the subscription model achieves positive cost savings over the benchmark dedicated only case in about a year, whereas the purchase model achieves cost savings in just over two years.

INSIGHT 9: Deploying cobots to augment dedicated pickers is an economically viable solution whether the store utilizes a subscription based model or purchases the cobots.

7. Conclusions and Future Research Directions

To address the challenges associated with the increasing demand for online grocery services, we proposed a dynamic order-picking policy where cobots synchronize with stochastically arriving in-store customers to augment the picking tasks of manual dedicated pickers. After modeling the problem as a Markov Decision Process, we designed a heuristic solution framework that a store could use to decide on alternatives at different stages of the decision-making process. Computational experiments using actual online grocery order and empirical shopping behavior data illustrates the feasibility of such a policy to achieve similar picking performance as the status quo (which is to deploy a limited set of dedicated pickers). However, performance varies based on the resource mix, initialization, abandon policy, updating policies, and environmental factors like participating in-store customers. Cobot assignments where waiting time per pick is arranged in an ascending manner result in higher pick rates than ones that create initial routes based on minimizing total travel time. Dynamic updates of resource allocation at pick completion events can help the store to achieve higher picking performance. While the cobot abandon policy insignificantly improves performance for ranking-based initially sequenced variants, such a policy has a higher positive impact when combined with TSP-based initial sequencing policy followed by dynamic reallocation policies. Similarly, exchanging only a single picking point among resources provides higher performance for ranking-based initial sequencing, yet, allowing multiple points shows better outcomes for TSP-based initial sequencing policies. Additionally, when there are high participation rate from in-store customers, ranking-based initial sequencing policies are preferred. Such policies are found to be computationally tractable for practical purposes, and can help the store to offload a significant amount of tasks to cobots. We find that three cobots and one dedicated picker can perform equivalently as two dedicated pickers, and this resource mix is economically viable.

As the first research to study order-picking policies using cobots and in-store customers in a retail store, there are numerous future research directions. First, we considered a wave-based policy, future research can be directed towards policies in wave-less systems, as well as investigating policies that consider multiple consecutive waves. More research is needed to better understand in-store customers' interests to participate, as well as their behavior, and performance in the proposed system. Lastly, this work considered the layout and facility design to be fixed; future research could jointly optimize other decisions, such as inventory, store shelf design, and item allocation, or even re-design the facility to more efficiently deploy such policies.

References

Ardjmand, E., H. Shakeri, M. Singh, and O. S. Bajgiran (2018). Minimizing order picking makespan with multiple pickers in a wave picking warehouse. *International Journal of Production Economics 206*, 169–183.

Aull, B., B. Coggins, S. Kohli, and E. Marohn (2022). The state of grocery in north america. *McKinsey Company*.

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

Azadeh, K., D. Roy, and M. de Koster (2020). Dynamic human-robot collaborative picking strategies. *Available at SSRN 3585396*.

Barcoding Inc. (2019). Agv best safety practices. https://www.fredagv.com/blog/agv-best-safety-practices.

Bayram, A. and B. Cesaret (2021). Order fulfillment policies for ship-from-store implementation in omni-channel retailing. *European Journal of Operational Research 294*(3), 987–1002.

Begley, S., B. Hancock, T. Kilroy, and S. Kohli (2019). Automation in retail: An executive overview for getting ready. *McKinsey & Company Retail Insights*.

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

Brick Meets Clicks (2020, Mar). Online grocery shopping surges to record levels in us during covid-19 crisis. https://www.foodlogistics.com/transportation/press-release/21125307/brick-meets-click-online-grocery-shopping-surges-to-record-levels-in-us-during-covid19-crisis.

Brittain Ladd (2023, Jun). Micro-fulfillment and the changing needs of grocery retailers. https://www.forbes.com/sites/forbescommunicationscouncil/2023/06/28/micro-fulfillment-and-the-changing-needs-of-grocery-retailers.

Bukchin, Y., E. Khmelnitsky, and P. Yakuel (2012). Optimizing a dynamic order-picking process. *European Journal of Operational Research 219*(2), 335–346.

Bureau of Labor Statistics (2023). Employer costs for employee compensation – june 2023. https://www.bls.gov/news.release/pdf/ecec.pdf.

Caporaso, T., D. Panariello, S. Grazioso, G. Di Gironimo, and L. Villani (2022). Robots helping humans: Collaborative shelf refilling. In *Robotics for Intralogistics in Supermarkets and Retail Stores*, pp. 117–135. Springer.

Chen, L., R. Jin, H. Qin, D. Simchi-Levi, and Z. Zhang (2019). Distributionally robust omnichannel stocking decisions in quick fulfillment systems. *Available at SSRN 3383881*.

Christensen, H., N. Amato, H. Yanco, M. Mataric, H. Choset, A. Drobnis, K. Goldberg, J. Grizzle, G. Hager, J. Hollerbach, et al. (2021). A roadmap for us robotics–from internet to robotics 2020 edition. *Foundations and Trends® in Robotics 8*(4), 307–424.

Clarke, G. and J. W. Wright (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations research 12*(4), 568–581.

Correll, N., K. E. Bekris, D. Berenson, O. Brock, A. Causo, K. Hauser, K. Okada, A. Rodriguez, J. M. Romano, and P. R. Wurman (2016). Analysis and observations from the first amazon picking challenge. *IEEE Transactions on Automation Science and Engineering 15*(1), 172–188.

Das, S., R. Ravi, and S. Sridhar (2023). Order fulfillment under pick failure in omnichannel ship-from-store programs. *Manufacturing & Service Operations Management 25*(2), 508–523.

Dayarian, I. and J. Pazour (2022). Crowdsourced order-fulfillment policies using in-store customers. *Production and Operations Management*.

Dethlefs, C., M. Ostermeier, and A. Hübner (2022). Rapid fulfillment of online orders in omnichannel grocery retailing. *EURO Journal on Transportation and Logistics 11*, 100082.

Difrancesco, R. M., I. M. van Schilt, and M. Winkenbach (2021). Optimal in-store fulfillment policies for online orders in an omni-channel retail environment. *European Journal of Operational Research 293*(3), 1058–1076.

Drew Holler (2021). Emphasizing opportunity, walmart moves more associates to full time. `https://corporate.walmart.com/news/2021/04/14/emphasizing-opportunity-walmart-moves-more-associates-to-full-time`.

Eriksson, E., A. Norrman, and J. Kembro (2022). Understanding the transformation toward omnichannel logistics in grocery retail: a dynamic capabilities perspective. *International Journal of Retail & Distribution Management 50*(8/9), 1095–1128.

Fager, P., F. Sgarbossa, and M. Calzavara (2021). Cost modelling of onboard cobot-supported item sorting in a picking system. *International Journal of Production Research 59*(11), 3269–3284.

Forrester Research (2019). The total economic impact™ of locus robotics. `https://locusrobotics.com/wp-content/uploads/Total-Economic-Impact-of-Locus-Robotics_NOV_2019F.pdf`.

Fragapane, G., R. de Koster, F. Sgarbossa, and J. O. Strandhagen (2021). Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda. *European Journal of Operational Research*.

Franklin, C. S., E. G. Dominguez, J. D. Fryman, and M. L. Lewandowski (2020). Collaborative robotics: New era of human–robot cooperation in the workplace. *Journal of Safety Research 74*, 153–160.

Gallino, S. and A. Moreno (2014). Integration of online and offline channels in retail: The impact of sharing reliable inventory availability information. *Management Science 60*(6), 1434–1451.

Gao, F. and X. Su (2017). Omnichannel retail operations with buy-online-and-pick-up-in-store. *Management Science 63*(8), 2478–2492.

Ghelichi, Z. and S. Kilaru (2021). Analytical models for collaborative autonomous mobile robot solutions in fulfillment centers. *Applied Mathematical Modelling 91*, 438–457.

Gong, Y. and R. De Koster (2008). A polling-based dynamic order picking system for online retailers. *IIE transactions 40*(11), 1070–1082.

Han, C., H. Jeon, J. Oh, and H. Lee (2022). Dynamic order picking method for multi-uav system in intelligent warehouse. *Remote Sensing 14*(23), 6106.

He, B., P. Mirchandani, and Y. Wang (2020). Removing barriers for grocery stores: O2o platform and self-scheduling delivery capacity. *Transportation Research Part E: Logistics and Transportation Review 141*, 102036.

He, Y., Q. Xu, and Z. Shao (2021). "ship-from-store" strategy in platform retailing. *Transportation Research Part E: Logistics and Transportation Review 145*, 102153.

Hübner, A., J. Hense, and C. Dethlefs (2022). The revival of retail stores via omnichannel operations: A literature review and research framework. *European Journal of Operational Research 302*(3), 799–818.

Hui, S. K., E. T. Bradlow, and P. S. Fader (2009). Testing behavioral hypotheses using an integrated model of grocery store shopping path and purchase behavior. *Journal of consumer research 36*(3), 478–493.

Instacart (2017). The instacart online grocery shopping dataset 2017. `https://www.instacart.com/datasets/grocery-shopping-2017`. Accessed: 2021-01-27.

Jacob, F., E. H. Grosse, S. Morana, and C. J. König (2023). Picking with a robot colleague: a systematic literature review and evaluation of technology acceptance in human–robot collaborative warehouses. *Computers & Industrial Engineering*, 109262.

Jin, M., G. Li, and T. C. E. Cheng (2018). Buy online and pick up in-store: Design of the service area. *European Journal of Operational Research 268*(2), 613–623.

Kelly Tyko (2021). Walmart to increase hours saturday with stores opening at 6 a.m.; senior hours to continue. `https://www.usatoday.com/story/money/shopping/2021/06/01/walmart-hours-2021-change-covid-senior-hours-continue-masks-update/5284374001/`.

Kraft, D. (1988). A software package for sequential quadratic programming. *Forschungsbericht- Deutsche Forschungs- und Versuchsanstalt fur Luft- und Raumfahrt*.

Lee, H.-Y. and C. C. Murray (2019). Robotics in order picking: evaluating warehouse layouts for pick, place, and transport vehicle routing systems. *International Journal of Production Research 57*(18), 5821–5841.

Li, R. (2020). Reinvent retail supply chain: Ship-from-store-to-store. *Production and Operations Management 29*(8), 1825–1836.

Liang, J., Z. Wu, C. Zhu, and Z.-H. Zhang (2020). An estimation distribution algorithm for wave-picking warehouse management. *Journal of Intelligent Manufacturing*, 1–14.

Lin, X., Y.-W. Zhou, and R. Hou (2021). Impact of a "buy-online-and-pickup-in-store" channel on price and quality decisions in a supply chain. *European Journal of Operational Research 294*(3), 922–935.

Liu, Y., Y. Xiao, and Y. Dai (2023). Omnichannel retailing with different order fulfillment and return options. *International Journal of Production Research 61*(15), 5053–5074.

Locus Robotics (2023). Locus origin. `https://locusrobotics.com/products/locus-origin/`.

Löffler, M., N. Boysen, and M. Schneider (2021). Picker routing in agv-assisted order picking systems. *INFORMS Journal on Computing*.

Löffler, M., N. Boysen, and M. Schneider (2023). Human-robot cooperation: Coordinating autonomous mobile robots and human order pickers. *Transportation Science*.

Lorson, F., A. Fügener, and A. Hübner (2023). New team mates in the warehouse: Human interactions with automated and robotized systems. *IISE Transactions 55*(5), 536–553.

Lu, W., D. McFarlane, V. Giannikas, and Q. Zhang (2016). An algorithm for dynamic order-picking in warehouse operations. *European Journal of Operational Research 248*(1), 107–122.

Lucas Systems (2020). The roi of autonomous mobile robots in your dc. https://www.lucasware.com/the-roi-of-autonomous-mobile-robots-in-your-dc/.

MacCarthy, B. L., L. Zhang, and L. Muyldermans (2019). Best performance frontiers for buy-online-pickup-in-store order fulfilment. *International Journal of Production Economics 211*, 251–264.

Marc Wulfraat (2023). Emphasizing opportunity, walmart moves more associates to full time. https://mwpvl.com/html/locus$_r obotics_{-i} ndependent_c onsultant_r eview.html$.

Masel, D. T. and A. Mesa (2018). Managing the order picking process in for click and collect in grocery stores.

Mayumi Brewster (2022). Annual retail trade survey shows impact of online shopping on retail sales during covid-19 pandemic. https://www.census.gov/library/stories/2022/04/ecommerce-sales-surged-during-pandemic.html.

McKinsey (2022, May). Achieving profitable online grocery order fulfillment. https://www.mckinsey.com/industries/retail/our-insights/achieving-profitable-online-grocery-order-fulfillment.

Meller, R. D., D. Nazzal, and L. M. Thomas (2018). Collaborative bots in distribution centers.

Mou, S. (2022a). In-store order fulfilment in omni-channel supermarkets with heterogeneous workforce: A bi-objective optimisation approach. *Computers & Industrial Engineering 171*, 108516.

Mou, S. (2022b). Integrated order picking and multi-skilled picker scheduling in omni-channel retail stores. *Mathematics 10*(9), 1484.

Nascimento, M. C., F. M. Toledo, and A. C. de Carvalho (2010). Investigation of a new grasp-based clustering algorithm applied to biological data. *Computers & Operations Research 37*(8), 1381–1388.

Neves-Moreira, F. and P. Amorim (2023). Playing hide and seek: tackling in-store picking operations while improving customer experience. *arXiv preprint arXiv:2301.02142*.

Ni, M., Q. He, X. Liu, and A. Hampapur (2019). Same-day delivery with crowdshipping and store fulfillment in daily operations. *Transportation Research Procedia 38*, 894–913.

Pasparakis, A., J. de Vries, and M. de Koster (2021). In control or under control? human-robot collaboration in warehouse order picking. *Human-Robot Collaboration in Warehouse Order Picking (March 31, 2021)*.

Pietri, N. O., X. Chou, D. Loske, M. Klumpp, and R. Montemanni (2021). The buy-online-pick-up-in-store retailing model: Optimization strategies for in-store picking and packing. *Algorithms 14*(12), 350.

Puterman, M. L. (2014). *Markov decision processes: discrete stochastic dynamic programming.* John Wiley & Sons.

Rasmi, S. A. B., Y. Wang, and H. Charkhgard (2022). Wave order picking under the mixed-shelves storage strategy: A solution method and advantages. *Computers & Operations Research 137*, 105556.

Repko, M. (2020). Grocers urgently need to fix broken online business model, as pandemic shifts more to web, report says. *CNBC*. https://www.cnbc.com/2020/07/09/coronavirus-grocers-must-fix-broken-online-model-report-says.html.

Roy Maurer (2023). Labor shortages forecast to persist for years. https://www.shrm.org/resourcesandtools/hr-topics/talent-acquisition/pages/labor-shortages-forecast-to-persist-2023.aspx.

Sağlam, B., F. S. Salman, S. Sayın, and M. Türkay (2006). A mixed-integer programming approach to the clustering problem with an application in customer segmentation. *European Journal of Operational Research 173*(3), 866–879.

Sahin, F. E. (2019). Open-source optimization algorithms for optical design. *Optik 178*, 1016–1022.

Salgado, T. M. F. D. N. (2015). In-store order picking routing: a biased random-key genetic algorithm approach.

Santini, A., C. E. Plum, and S. Ropke (2018). A branch-and-price approach to the feeder network design problem. *European Journal of Operational Research 264*(2), 607–622.

Seghezzi, A., C. Siragusa, and R. Mangiaracina (2022). Enhancing in-store picking for e-grocery: an empirical-based model. *International Journal of Physical Distribution & Logistics Management 52*(4), 301–323.

Shen, H., F. Namdarpour, and J. Lin (2022). Investigation of online grocery shopping and delivery preference before, during, and after covid-19. *Transportation Research Interdisciplinary Perspectives 14*, 100580.

Srinivas, S. and S. Yu (2022). Collaborative order picking with multiple pickers and robots: Integrated approach for order batching, sequencing and picker-robot routing. *International Journal of Production Economics 254*, 108634.

Stephanie Ferguson (2023). Understanding america's labor shortage: The most impacted industries. https://www.uschamber.com/workforce/understanding-americas-labor-shortage-the-most-impacted-industries.

Thomas Ozbun (2023). Number of weekly customer visits to walmart stores worldwide from fiscal year 2017 to 2023. https://www.statista.com/statistics/818929/number-of-weekly-customer-visits-to-walmart-stores-worldwide/ .

Tilk, C., A.-K. Rothenbächer, T. Gschwind, and S. Irnich (2017). Asymmetry matters: Dynamic half-way points in bidirectional labeling for solving shortest path problems with resource constraints faster. *European Journal of Operational Research 261*(2), 530–539.

Tompkins, J. A., J. A. White, Y. A. Bozer, and J. M. A. Tanchoco (2010). *Facilities planning*. John Wiley & Sons.

Ulmer, M. W., J. C. Goodson, D. C. Mattfeld, and B. W. Thomas (2020). On modeling stochastic dynamic vehicle routing problems. *EURO Journal on Transportation and Logistics 9*(2), 100008.

U.S. Bureau of Labor Statistics (2023). Employment cost index – june 2023. `https://www.bls.gov/news.release/pdf/eci.pdf`.

Vazquez-Noguerol, M., J. Comesaña-Benavides, R. Poler, and J. C. Prado-Prado (2022). An optimisation approach for the e-grocery order picking and delivery problem. *Central European Journal of Operations Research 30*(3), 961–990.

Walmart (2023a). How much do walmart associates make? `https://corporate.walmart.com/askwalmart/how-much-do-walmart-associates-make`.

Walmart (2023b). Location facts. `https://corporate.walmart.com/about/location-facts`.

Wang, H., W. Chen, and J. Wang (2020). Coupled task scheduling for heterogeneous multi-robot system of two robot types performing complex-schedule order fulfillment tasks. *Robotics and Autonomous Systems 131*, 103560.

Wang, K., Y. Li, and Y. Zhou (2022). Execution of omni-channel retailing based on a practical order fulfillment policy. *Journal of Theoretical and Applied Electronic Commerce Research 17*(3), 1185–1203.

Winkelhaus, S., M. Zhang, E. H. Grosse, and C. H. Glock (2022). Hybrid order picking: A simulation model of a joint manual and autonomous order picking system. *Computers & Industrial Engineering 167*, 107981.

Yang, D. and X. Zhang (2020). Omnichannel operations with ship-from-store. *Operations Research Letters 48*(3), 257–261.

Yokota, T. (2019). Min-max-strategy-based optimum co-operative picking with agvs in warehouse. In *2019 58th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, pp. 236–242. IEEE.

Yu, T., D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine (2020). Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pp. 1094–1100. PMLR.

Zhang, J., F. Liu, J. Tang, and Y. Li (2019). The online integrated order picking and delivery considering pickers' learning effects for an o2o community supermarket. *Transportation Research Part E: Logistics and Transportation Review 123*, 180–199.

Zhang, M., E. H. Grosse, and C. H. Glock (2023). Ergonomic and economic evaluation of a collaborative hybrid order picking system. *International Journal of Production Economics 258*, 108774.

Zhang, M., S. Winkelhaus, and E. H. Grosse (2021). Evaluation of human workload in a hybrid order picking system. *IFAC-PapersOnLine 54*(1), 458–463.

Zhang, N. and J. Pazour (2019). Expected travel distance models for retail store order fulfillment. In *IIE Annual Conference. Proceedings*, pp. 94–99. Institute of Industrial and Systems Engineers (IISE).

Zhu, S., H. Wang, X. Zhang, X. He, and Z. Tan (2022). A decision model on human-robot collaborative routing for automatic logistics. *Advanced Engineering Informatics 53*, 101681.

Zou, Y., D. Zhang, and M. Qi (2019). Order picking system optimization based on picker-robot collaboration. In *Proceedings of the 2019 5th International Conference on Industrial and Business Engineering*, pp. 1–6.

Žulj, I., H. Salewski, D. Goeke, and M. Schneider (2022). Order batching and batch sequencing in an amr-assisted picker-to-parts system. *European Journal of Operational Research 298*(1), 182–201.

Appendix A: Categorizing and Defining Transition Probabilities

We categorize the transition probabilities into two types. One is to transition to a pick completion state s_{e+1} by reaching either τ_k^c or τ_k^d. The second type is to transition to a non-pick completion state s_{e+1} by τ_k^r or τ_k^a. The first type transitions with the probability that a specific picking resource $k \in K$ completes the required number of picks (n_v) at their currently assigned stopping point before the other deployed picking resources. For any $\gamma \in K$ this probability is calculated from three-time components: (1) travel time from the current stopping point location, ∂_γ, to the next assigned stopping point location, θ_γ, denoted as $t^C_{\partial\theta} \vee t^D_{\partial\theta}$, (2) uncertain waiting time at θ_γ denoted as $\bar{\omega}_{\gamma\theta}$, and (3) uncertain picking time at θ_γ denoted as $\bar{s}_{\gamma\theta}$. The travel time component can be of value 0 if any $\gamma \in K$ in the system has already reached θ_γ. Similarly, if synchronization with an in-store customer at θ_γ has already occurred for a cobot, $\bar{\omega}_{\gamma\theta} = 0$. However, to define the probability of pick completion of $\gamma \in K$, we must also take into account the probability of any τ_k^r or a τ_k^a happening before the pick completion of $\gamma \in K$. Thus, the transition probability at decision epoch e of a picking resource $\gamma \in K$ completing n_v picks at θ_γ before the other picking resources pick completion and other triggers can be defined simply by (44) and in detail by (45) for any $k \in C$ and (46) for any $k \in D$.

$$\begin{aligned}\delta^p_{\gamma e} = \text{probability } \Big[&(\text{pick completion by } \gamma \in K \text{ before pick completion by } k \in K \setminus \gamma),\\ &(\text{pick completion by } \gamma \in K \text{ before wave-time trigger by any } k \in K),\\ &(\text{pick completion by } \gamma \in K \text{ before abandon trigger by any } k \in K)\Big]\end{aligned} \tag{44}$$

$$\begin{aligned}\delta^{pC}_{\gamma e} = \text{probability}\Big[&(t^C_{\partial\theta} + \bar{\omega}_{\gamma\theta} + \bar{s}_\theta + t_e) \textit{ for } \gamma \in C < (t^C_{\partial\theta} + \bar{\omega}_{\gamma\theta} + \bar{s}_\theta + t_e)\forall k \in C \setminus \{\gamma\},\\ &(t^C_{\partial\theta} + \bar{\omega}_{\gamma\theta} + \bar{s}_\theta + t_e) \textit{ for } \gamma \in C < (t^D_{\partial\theta} + \bar{\omega}_{\gamma\theta} + \bar{s}_\theta + t_e)\forall k \in D,\\ &(t^C_{\partial\theta} + \bar{\omega}_{\gamma\theta} + \bar{s}_\theta + t_e) \textit{ for } \gamma \in C < (T - t^C_{\theta p})\forall k \in C,\\ &(t^C_{\partial\theta} + \bar{\omega}_{\gamma\theta} + \bar{s}_\theta + t_e) \textit{ for } \gamma \in C < (T - t^D_{\theta p})\forall k \in D,\\ &(t^D_{\partial\theta} + \bar{\omega}_{\gamma\theta} + \bar{s}_\theta + t_e) \textit{ for } \gamma \in D < (t_e | \tau^u = \tau^a_k)\forall k \in K\Big]\end{aligned} \tag{45}$$

$$\delta^{pD}_{\gamma e} = \text{probability}\Big[(t^D_{\partial\theta} + \bar{\omega}_{\gamma\theta} + \bar{s}_\theta + t_e) \ for\ \gamma \in D < (t^D_{\partial\theta} + \bar{\omega}_{\gamma\theta} + \bar{s}_\theta + t_e)\forall k \in D \setminus \{\gamma\},$$
$$(t^D_{\partial\theta} + \bar{\omega}_{\gamma\theta} + \bar{s}_\theta + t_e) \ for\ \gamma \in D < (t^D_{\partial\theta} + \bar{\omega}_{\gamma\theta} + \bar{s}_\theta + t_e)\forall k \in C,$$
$$(t^D_{\partial\theta} + \bar{\omega}_{\gamma\theta} + \bar{s}_\theta + t_e) \ for\ \gamma \in D < (T - t^C_{\theta p})\forall k \in C, \tag{46}$$
$$(t^D_{\partial\theta} + \bar{\omega}_{\gamma\theta} + \bar{s}_\theta + t_e) \ for\ \gamma \in D < (T - t^D_{\theta p})\forall k \in D,$$
$$(t^D_{\partial\theta} + \bar{\omega}_{\gamma\theta} + \bar{s}_\theta + t_e) \ for\ \gamma \in D < (t_e|\tau^u = \tau^a_k)\forall k \in K\Big]$$

The second type is the probability of reaching the wave-time trigger τ^r_γ of picking resource $\gamma \in K$ before any other any other $\tau^r_k \forall k \in K \setminus \{\gamma\}$, any pick completion trigger, and any abandon trigger. This can be simplified by finding out the probability of the earliest wave-time trigger happening before any resource completes a pick and before any τ^a. In other words, it would be the probability of reaching state s_{e+1} from state s_e, such that the remaining time in the wave $(T - t_{e+1})$ would be equal to the maximum of the travel time from current stopping point ∂_γ to the dropoff station $v_p \forall k \in K$. Hence, this probability can be defined by equation (47).

$$\delta^r_e = \text{probability}\Big[T - max\ (t^C_{\theta p} \quad \forall k \in C, t^D_{\theta p} \quad \forall k \in D) < (t^C_{\theta p} + \bar{\omega}_{k\theta} + \bar{s}_\theta + t_e) \quad \forall k \in C,$$
$$T - max\ (t^C_{\theta p} \quad \forall k \in C, t^D_{\theta p} \quad \forall k \in D) < (t^D_{\theta p} + \bar{\omega}_{k\theta} + \bar{s}_\theta + t_e) \quad \forall k \in D, \tag{47}$$
$$T - max\ (t^C_{\theta p} \quad \forall k \in C, t^D_{\theta p} \quad \forall k \in D) < (t_e|\tau^u = \tau^a_k)\forall k \in K\Big]$$

Lastly, the transition probability for having a triggering event of type τ^a_k for resource $\gamma \in K$ can also be defined in a similar manner by (48)

$$\delta^a_{\gamma e} = \text{probability}\Big[(t_e|\tau^u = \tau^a_\gamma) < (t^C_{\theta p} + \bar{\omega}_{k\theta} + \bar{s}_\theta + t_e) \quad \forall k \in C,$$
$$(t_e|\tau^u = \tau^a_\gamma) < (t^D_{\theta p} + \bar{\omega}_{k\theta} + \bar{s}_\theta + t_e) \quad \forall k \in D,$$
$$(t_e|\tau^u = \tau^a_\gamma) < T - max\ (t^C_{\theta p} \quad \forall k \in C, t^D_{\theta p} \quad \forall k \in D), \tag{48}$$
$$(t_e|\tau^u = \tau^a_\gamma) < (t_e|\tau^u = \tau^a_k)\forall k \in K \setminus \{\gamma\}\Big]$$

Appendix B: Picking performance across instances

Instances/K	C1D1	C2D1	C3D1	C4D1	C0D1	C0D2
1	95.79%	88.51%	99.33%	99.28%	75.00%	100.00%
2	78.66%	99.44%	99.65%	99.74%	61.11%	100.00%
3	84.53%	92.64%	98.21%	98.68%	62.96%	95.06%
4	82.26%	92.66%	96.55%	98.02%	64.71%	95.29%
5	79.69%	91.09%	96.88%	98.20%	49.44%	94.38%
6	70.42%	82.85%	94.54%	97.72%	60.00%	92.00%
7	72.88%	75.27%	85.77%	93.38%	44.34%	84.91%
8	74.67%	85.78%	93.48%	96.58%	54.70%	85.47%
9	64.27%	72.74%	81.37%	89.08%	44.00%	74.40%
10	62.72%	70.96%	81.81%	87.23%	41.35%	74.44%

Table 4 Picking performance across instances for increasing cobots with one dedicated picker compared to benchmark resource mix

Appendix C: Stopping Point to Product Category Mapping

Stopping Point Number	Product Category	Stopping Point Number	Product Category	Stopping Point Number	Product Category
1	prepared soups salads	46	mint gum	91	soy lactosefree
2	specialty cheeses	47	vitamins supplements	92	baby food formula
3	energy granola bars	48	breakfast bars pastries	93	breakfast bakery
4	instant foods	49	packaged poultry	94	tea
5	marinades meat preparation	50	fruit vegetable snacks	95	canned meat seafood
6	other	51	preserved dips spreads	96	lunch meat
7	packaged meat	52	frozen breakfast	97	baking supplies decor
8	bakery desserts	53	cream	98	juice nectars
9	pasta sauce	54	paper goods	99	canned fruit applesauce
10	kitchen supplies	55	shave needs	100	missing
11	cold flu allergy	56	diapers wipes	101	air fresheners candles
12	fresh pasta	57	granola	102	baby bath body care
13	prepared meals	58	frozen breads doughs	103	ice cream toppings
14	tofu meat alternatives	59	canned meals beans	104	spices seasonings
15	packaged seafood	60	trash bags liners	105	doughs gelatins bake mixes
16	fresh herbs	61	cookies cakes	106	hot dogs bacon sausage
17	baking ingredients	62	white wines	107	chips pretzels
18	bulk dried fruits vegetables	63	grains rice dried goods	108	other creams cheeses
19	oils vinegars	64	energy sports drinks	109	skin care
20	oral hygiene	65	protein meal replacements	110	pickled goods olives
21	packaged cheese	66	asian foods	111	plates bowls cups flatware
22	hair care	67	fresh dips tapenades	112	bread
23	popcorn jerky	68	bulk grains rice dried goods	113	frozen juice
24	fresh fruits	69	soup broth bouillon	114	cleaning products
25	soap	70	digestion	115	water seltzer sparkling water
26	coffee	71	refrigerated pudding desserts	116	frozen produce
27	beers coolers	72	condiments	117	nuts seeds dried fruit
28	red wines	73	facial care	118	first aid
29	honeys syrups nectars	74	dish detergents	119	frozen dessert
30	latino foods	75	laundry	120	yogurt
31	refrigerated	76	indian foods	121	cereal
32	packaged produce	77	soft drinks	122	meat counter
33	kosher foods	78	crackers	123	packaged vegetables fruits
34	frozen meat seafood	79	frozen pizza	124	spirits
35	poultry counter	80	deodorants	125	trail mix snack mix
36	butter	81	canned jarred vegetables	126	feminine care
37	ice cream ice	82	baby accessories	127	body lotions soap
38	frozen meals	83	fresh vegetables	128	tortillas flat bread
39	seafood counter	84	milk	129	frozen appetizers sides
40	dog food care	85	food storage	130	hot cereal pancake mixes
41	cat food care	86	eggs	131	dry pasta
42	frozen vegan vegetarian	87	more household	132	beauty
43	buns rolls	88	spreads	133	muscles joints pain relief
44	eye ear care	89	salad dressing toppings	134	specialty wines champagnes
45	candy chocolate	90	cocoa drink mixes		

Table 5 Stopping point number and its mapped product category

Appendix D: Order Profiles

Instance No	Order Profile - (Stopping Point Number, Number of items to pick)	N	$\lvert V^r \rvert$
1	(1, 1), (4, 1), (16, 1), (19, 1), (21, 2), (24, 7), (26, 4), (32, 1), (37, 1), (45, 1), (54, 2), (57, 2), (64, 1), (74, 1), (78, 3), (81, 1), (83, 7), (86, 1), (93, 1), (98, 3), (100, 1), (101, 1), (108, 1), (115, 5), (120, 3), (121, 1), (123, 4), (127, 1), (129, 1)	60	30
2	(16, 2), (19, 1), (20, 1), (21, 3), (24, 7), (26, 2), (31, 3), (36, 1), (37, 2), (42, 1), (53, 1), (75, 1), (77, 3), (78, 1), (83, 11), (84, 4), (86, 4), (107, 1), (109, 1), (110, 1), (112, 2), (114, 1), (115, 2), (120, 4), (121, 2), (123, 6), (128, 2), (129, 2)	72	29
3	(3, 1), (4, 2), (16, 1), (19, 4), (21, 4), (24, 6), (31, 1), (36, 2), (37, 2), (45, 1), (50, 1), (54, 1), (59, 2), (61, 1), (67, 2), (69, 1), (72, 1), (77, 2), (78, 1), (79, 2), (83, 6), (84, 5), (86, 1), (88, 1), (91, 1), (93, 4), (94, 1), (96, 4), (100, 4), (107, 2), (108, 2), (110, 1), (112, 5), (116, 1), (120, 1), (121, 1), (123, 2), (127, 1)	81	39
4	(4, 1), (6, 1), (9, 1), (12, 2), (16, 2), (19, 1), (21, 1), (24, 12), (30, 2), (32, 2), (36, 1), (37, 1), (45, 1), (48, 1), (51, 1), (52, 1), (54, 1), (56, 1), (66, 1), (70, 1), (74, 1), (82, 1), (83, 10), (84, 4), (85, 1), (86, 2), (87, 1), (91, 4), (96, 1), (98, 2), (100, 2), (104, 3), (106, 5), (110, 1), (114, 2), (115, 1), (116, 1), (120, 2), (123, 5), (129, 1)	85	41
5	(3, 3), (16, 2), (17, 1), (19, 1), (20, 1), (21, 5), (24, 13), (26, 1), (31, 1), (35, 1), (36, 1), (37, 1), (48, 2), (60, 1), (62, 1), (67, 1), (69, 2), (72, 1), (77, 1), (78, 2), (81, 3), (83, 8), (84, 2), (86, 1), (91, 1), (92, 1), (93, 1), (94, 4), (96, 1), (98, 2), (99, 4), (100, 1), (104, 1), (107, 2), (108, 1), (112, 1), (115, 1), (116, 1), (117, 1), (120, 5), (122, 1), (123, 2), (128, 1), (131, 1)	89	45
6	(2, 2), (9, 1), (11, 1), (16, 4), (17, 2), (19, 2), (21, 4), (24, 6), (25, 1), (31, 3), (34, 1), (36, 1), (37, 1), (53, 1), (54, 1), (59, 4), (63, 1), (67, 1), (69, 1), (71, 1), (72, 1), (74, 1), (75, 1), (78, 1), (81, 2), (83, 12), (84, 2), (85, 1), (86, 1), (91, 2), (95, 1), (96, 4), (98, 1), (99, 1), (106, 1), (107, 3), (108, 1), (111, 1), (112, 1), (115, 2), (116, 3), (117, 1), (120, 3), (123, 11), (128, 1), (129, 1), (133, 1)	100	48
7	(1, 1), (4, 1), (6, 1), (9, 2), (19, 2), (21, 2), (24, 15), (25, 1), (26, 1), (31, 3), (32, 1), (34, 1), (35, 1), (37, 1), (38, 1), (43, 1), (45, 1), (50, 3), (52, 2), (53, 3), (54, 1), (61, 2), (62, 1), (66, 2), (67, 1), (72, 1), (75, 1), (78, 1), (79, 1), (80, 1), (83, 8), (84, 4), (86, 1), (91, 1), (93, 2), (98, 2), (99, 1), (104, 1), (105, 3), (106, 2), (107, 2), (108, 1), (112, 2), (115, 5), (120, 3), (121, 1), (123, 5), (125, 1), (129, 2), (130, 1), (131, 2)	106	52
8	(1, 1), (4, 1), (9, 2), (13, 2), (16, 2), (23, 1), (24, 11), (25, 1), (31, 3), (34, 1), (36, 2), (37, 4), (38, 1), (42, 1), (45, 2), (48, 2), (50, 1), (53, 2), (59, 1), (61, 1), (64, 1), (67, 2), (69, 3), (77, 5), (78, 2), (79, 1), (81, 3), (83, 20), (84, 6), (86, 1), (89, 1), (94, 2), (96, 1), (98, 1), (100, 1), (107, 4), (112, 1), (115, 6), (117, 3), (120, 4), (121, 1), (123, 4), (129, 1), (134, 1)	117	45
9	(3, 4), (4, 1), (5, 1), (8, 1), (9, 1), (20, 1), (21, 9), (23, 1), (24, 10), (26, 2), (31, 4), (32, 1), (33, 1), (37, 3), (38, 1), (42, 1), (45, 2), (51, 1), (53, 1), (54, 1), (59, 1), (61, 1), (63, 1), (65, 1), (66, 2), (69, 2), (74, 1), (77, 2), (78, 1), (83, 10), (84, 5), (85, 1), (86, 1), (89, 1), (91, 5), (94, 1), (95, 1), (98, 1), (106, 2), (107, 1), (108, 2), (112, 3), (115, 4), (116, 1), (120, 8), (121, 4), (123, 11), (129, 3), (131, 1)	125	50
10	(2, 2), (3, 7), (5, 2), (12, 1), (16, 1), (17, 2), (20, 1), (21, 3), (23, 1), (24, 11), (26, 2), (30, 2), (32, 1), (34, 1), (36, 1), (37, 2), (42, 1), (45, 3), (46, 1), (50, 1), (53, 2), (54, 2), (57, 1), (61, 1), (66, 1), (69, 3), (74, 1), (77, 4), (78, 1), (81, 4), (83, 11), (84, 2), (88, 3), (91, 3), (92, 1), (93, 1), (94, 1), (96, 2), (98, 1), (106, 1), (107, 9), (108, 3), (110, 2), (115, 2), (116, 1), (117, 5), (120, 10), (123, 4), (125, 1), (128, 2), (130, 1), (131, 1)	133	53

Table 6 Generated order instances from whole orders (Instacart 2017)