information

MDPI

Article

Optimization of Worker Redeployment for Enhancing Customer Service Performance

Hyunho Kim [1], Wonseok Kang [2,*] and Eunmi Lee [3]

[1] Department of Economics, Korea Military Academy, Seoul 01805, Republic of Korea; kimhyuh@kma.ac.kr
[2] Department of Management, Korea Military Academy, Seoul 01805, Republic of Korea
[3] College of General Education, Kookmin University, Seoul 02707, Republic of Korea; foodbo@kookmin.ac.kr
* Correspondence: kws0125@kma.ac.kr

Abstract: This study considers ways in which workers can be allocated dynamically in the few hours before the truck departure to flush the system of orders that are almost completed, thereby increasing the service performance of the system. To implement a worker allocation policy correctly, we need to answer the following questions: how many workers should we move, and when? We present the optimal number of workers required and the switching time for the proposed three dynamic worker reallocation policies through simulation experiments. The number of workers required was determined by the difference between the current and target probability of success of an order in the system based on the state-dependent sojourn time distribution, and the performance of the system was measured by Next Scheduled Departure (NSD). We find that the policies with late switching times and higher target probability of success have a greater effect on customer satisfaction. Our results suggest that it is possible to improve service performance significantly, in some conditions, by moving the right number of workers to the right place at the right time.

Keywords: order fulfillment system; dynamic worker allocation policy; NSD (next scheduled departure); probability of success; flushing policy; cascade policy

check for updates

Academic Editor: Vladimír Bureš

Received: 25 December 2024
Revised: 22 January 2025
Accepted: 11 February 2025
Published: 18 February 2025

Citation: Kim, H.; Kang, W.; Lee, E. Optimization of Worker Redeployment for Enhancing Customer Service Performance. *Information* **2025**, *16*, 149. https://doi.org/10.3390/info16020149

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Order fulfillment is the cornerstone of most distribution centers, as it represents the process of meeting customer demands. Typically, these centers focus on three core tasks: picking, packing, and shipping. Orders are received throughout the day and processed for shipment via the picking and packing stages. Customers often prioritize quick delivery, making the efficiency of responding to orders a critical concern. Orders that are ready to go to consumers immediately after the last delivery truck leaves the distribution center have to wait a long time for the next delivery truck. This raises the following question: could reallocating workers shortly before the truck's departure have expedited the completion of nearly finished orders, thereby enhancing service performance? For instance, reallocating workers from picking to shipping in the final hours might help to "flush" the system. However, this strategy risks disrupting system balance, temporarily reducing throughput in the picking area and potentially causing delays in subsequent operations. This paper examines whether such adjustments can effectively improve system performance.

The operational effectiveness of order fulfillment systems has long been a subject of academic inquiry. However, while studies have primarily concentrated on operational metrics such as throughput and cycle time [1], the customer-centric performance metric of Next Scheduled Departure (NSD) remains underexplored. NSD, which directly correlates with customer satisfaction, necessitates dynamic, real-time worker allocation policies

capable of responding to fluctuating system states. This research bridges this gap by developing and testing adaptive worker allocation strategies modeled on state-dependent sojourn time distributions. This study's conceptual model integrates principles of queuing theory with phase-type distribution and NSD, enabling actionable insights for practitioners in high-variability environments.

This study focuses on optimizing a performance metric known as Next Scheduled Departure (NSD), introduced by [2]. NSD quantifies the percentage of orders received within a 24h period between successive cutoff times that make it onto the truck departing on the same day. Cutoff times, set by distribution center managers, dictate which orders are due for shipment on a given day. According to [2], improvements in NSD correlate directly with increased customer satisfaction. Unlike traditional metrics such as work-in-progress (WIP), cycle time, or throughput—which prioritize operational efficiency—NSD emphasizes performance from the customer's perspective.

To explore dynamic worker allocation, we first consider a straightforward policy: moving a fixed number of workers at a set time near the daily deadline. This study models the order fulfillment process as a sequence of three workstations (picking, packing, and shipping) staffed with 10, 12, and 9 workers, respectively. In this scenario, eight workers are reallocated from picking to shipping one hour before the deadline each day to expedite incomplete orders. While occasionally effective, this static policy often fails to account for real-time system conditions, leading to inefficiencies. For example, workers might be transferred from an active picking station to an underutilized shipping station, resulting in idle time and bottlenecks.

These observations prompted the development of more advanced, context-aware strategies based on sojourn time. This approach evaluates the likelihood of an order meeting its shipping deadline without reallocating workers. When this probability falls below a predefined threshold, workers are reassigned to improve the chances of timely completion for high-priority orders. This paper introduces several dynamic worker allocation strategies designed to enhance customer satisfaction without compromising system stability. This study aims to answer two questions: First, is it possible to improve the performance of the system, i.e., consumer satisfaction as measured by NSD, by dynamic worker placement? Second, for a dynamic worker placement policy to be effective, when and how many workers should be placed?

The next section reviews prior research on dynamic worker allocation, followed by a discussion of key concepts like the NSD metric and probability of success. Finally, this study presents and evaluates various policies through simulations and concludes with insights and recommendations.

2. Literature Review

For our literature review, we employed a narrative review approach, a traditional method that provides a qualitative summary of relevant literature. We examined a wide range of literature on dynamic worker allocation, chronologically exploring from early concepts to recent advancements in predictive analytics and IoT-enabled sensors. The analysis focused on key themes and concepts such as different types of worker allocation strategies, optimization approaches, and performance metrics. Through this process, we identified the strengths, weaknesses, and limitations of various approaches by different authors and presented three key differentiators that our study aims to address.

A company's goal is to maximize consumer satisfaction by optimizing the balance of quality, cost, and delivery. To this end, production improvement activities such as Total Quality Control and Just In Time have been carried out mainly by Japanese companies, and Kaplan et al. [3] proposed the Balanced Scorecard model, which measures the performance

of a company from four perspectives—financial, consumer, internal, and learning and growth perspectives. Osterwalder et al. [4] reported that the performance of a company is connected to internal management and production activities and consumer-oriented activities such as delivery and after-sales service, and proposed a business model that diagnoses the current status and improves strengths and weaknesses. Among the various methods of improving and measuring the performance of a company, the study focused on maximizing performance from the consumer perspective by shortening the delivery time of orders through the company's production activities, that is, dynamic relocation of workers, and investigated related literature.

The study of dynamic task allocation for workers has been extensively explored in manufacturing, with some attention in warehousing. This research field spans several concepts, including work-sharing systems, cross-training or cross-utilization of workers, collaborative versus non-collaborative systems, and agile workforce strategies. Many researchers have investigated how cross-trained employees can enhance manufacturing system performance.

Askin et al. [5] categorized work-sharing strategies into Dynamic Assembly-Line Balancing (DLB) and Moving Worker Modules (MWM). In MWM systems, the number of workers is fewer than the machines, leading to shared task responsibilities across zones, whereas DLB involves equal numbers of machines and workers. DLB divides tasks into fixed and shared categories, with fixed tasks assigned to specific workers and shared tasks handled by adjacent pairs. The review classifies floating worker systems into MWM or DLB, analyzing their characteristics based on worker-to-machine ratios, skill levels, and work-in-progress (WIP).

Representative examples of MWM include the Toyota Sewn-products Management System (TSS) and the Bucket Brigade System, as described by [3]. TSS assigns workers to specific zones where they complete tasks at each machine downstream until either the task concludes or another worker takes over. Bischak [6] analyzed a U-shaped manufacturing line operating under TSS and demonstrated that such systems outperform fixed worker models, even without buffers. Similarly, Zavadlav et al. [7] utilized Markov decision processes and simulation to evaluate U-shaped serial lines under TSS, finding that free-floating worker assignments most effectively reduce WIP.

Bartholdi III et al. [8] expanded on TSS with their bucket brigades protocol, demonstrating that sequencing workers from slowest to fastest can naturally balance tasks and maximize production rates. However, McClain et al. [9] highlighted limitations in scenarios involving random task times or similar worker speeds, which could result in worker idleness. They proposed modifications, such as relaxing the "wait" rule and incorporating small inventories, to address these inefficiencies.

In the context of DLB systems, Gel et al. [10] proposed a zoning strategy for CONWIP production systems utilizing hierarchical cross-training. They introduced a "fixed-before-shared" policy, emphasizing that cross-trained workers should prioritize unique tasks before assisting with shared responsibilities. The approach was shown to significantly benefit system performance when cross-trained workers demonstrated higher efficiency than their static counterparts. Gel et al. [11] further identified factors, such as the ability to preempt tasks, task granularity, and reduced variability, that enhance work-sharing opportunities.

The Half Full Buffer (HFB) control policy is another notable concept, providing enough work for downstream workers while maintaining empty space to prevent upstream worker blockages. Research by [9,11,12] and others explored variations of this policy, showing how it increases system performance. Chen et al. [13] introduced the Smallest R No Starvation

(SRNS) rule, which calculates threshold values to optimize cross-training under CONWIP conditions, further confirming the effectiveness of HFB policies.

Several studies have explored optimizing tandem line performance with finite buffers, focusing on dynamic server allocation to achieve specific objectives. Andradottir et al. [14] examined tandem lines with flexible, heterogeneous servers to determine optimal dynamic assignments that maximize long-run average throughput. They found that when there is no trade-off between server synergy and specialization, the best approach involves servers working in teams of two or more. Tekin et al. [15] developed generalized round-robin policies for server assignments in queueing networks with demand exceeding service capacity, aiming to maximize throughput. Similarly, Isik et al. [16] studied dynamic server allocation in tandem queueing systems with non-collaborating servers, focusing on throughput optimization. Other studies have considered minimizing operational costs as a primary goal. Kırkızlar et al. [17] analyzed server assignments to maximize long-run average profit, incorporating holding costs into the decision-making process. Additionally, Andradottir et al. [14] proposed a dynamic server assignment policy where server movements are determined by the number of jobs, server locations, and threshold values, balancing system performance and cost efficiency. These approaches highlight the importance of flexible and dynamic server allocation in enhancing tandem line performance.

Distinct from tandem lines, Ganbold et al. [18] investigated dynamic worker reallocation within warehouses, presenting a simulation-based optimization method to enhance daily productivity. Their approach combined discrete-event simulation with random neighborhood search techniques.

Recent research has provided additional insights into dynamic worker allocation. For instance, a notable study by [19] integrated collaborative multi-agent systems to address complex scheduling problems in high-density environments, highlighting the role of cooperative dynamics in enhancing overall efficiency.

Further advancements include adaptive queuing models proposed by [20], which utilize Bayesian inference to dynamically adjust server allocation based on predictive demand patterns. Huang et al. [21] examined the impact of workforce cross-training on service quality, emphasizing the importance of versatility in worker skill sets for mitigating delays during peak periods. These studies collectively underscore the necessity for policies that are not only dynamic but also context-aware, leveraging advanced analytics and machine learning to achieve optimal results.

Expanding on these findings, Lam et al. [22] analyzed the integration of predictive analytics into workforce management, demonstrating a significant reduction in response times for high-priority tasks. Similarly, Wang et al. [23] highlighted the role of real-time digital twins in simulating allocation scenarios, providing managers with actionable insights for immediate decision-making. Chu et al. [24] further explored the implications of integrating IoT-enabled sensors in dynamic worker allocation, enabling data-driven adjustments based on real-time feedback from operational environments. Collectively, these advancements demonstrate a clear trajectory toward leveraging emerging technologies to refine task allocation strategies.

This paper introduces a unique dynamic worker allocation policy that addresses three key gaps in the literature: (1) While previous studies primarily aim to increase throughput, minimize cycle time, or reduce WIP, this research prioritizes customer service improvement in order fulfillment systems, measured via the Next Scheduled Departure (NSD) metric ([2]). (2) Unlike prior research focusing on small tandem lines (2–3 stations), this work targets larger, real-world systems with numerous workers across various stations like picking, packing, and shipping. (3) By leveraging state-dependent sojourn time distributions, this study computes probabilities for order success and dynamically reallocates workers to

maximize these probabilities, rather than relying solely on heuristic or simulation-based approaches.

3. Preliminaries

To establish the foundation for our worker allocation strategies, we first introduce three core concepts critical to our design: Next Scheduled Departure (NSD), state-dependent sojourn time distributions, and the probability of success.

3.1. Next Scheduled Departure (NSD)

According to [2], NSD represents the percentage of orders received within a 24 h timeframe, from one cutoff point to the next, that are successfully loaded onto the truck departing on the same day. Earlier cutoff times increase the likelihood of orders meeting the deadline, thereby elevating the NSD. Figure 1 demonstrates the correlation between cutoff times and NSD, highlighting the influence of the deadline on performance.

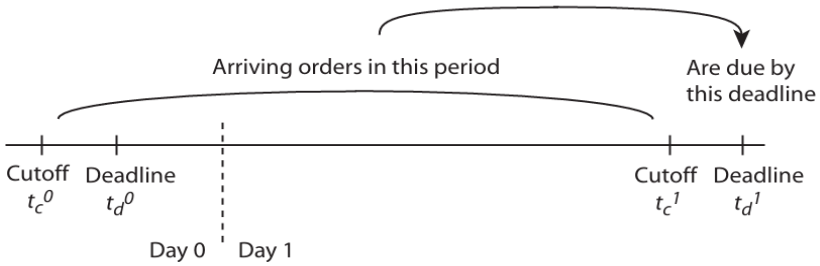

Figure 1. The connection between cutoff time and NSD [8].

The expected NSD can be derived from the steady-state sojourn time distribution. Orders arriving closer to the deadline have a lower probability of timely departure compared with those arriving earlier in the 24 h cycle. As depicted in Figure 2, the expected NSD corresponds to the average probability of timely completion for all orders during this period. This probability can be computed for multi-server queuing systems using the methods developed by [25].

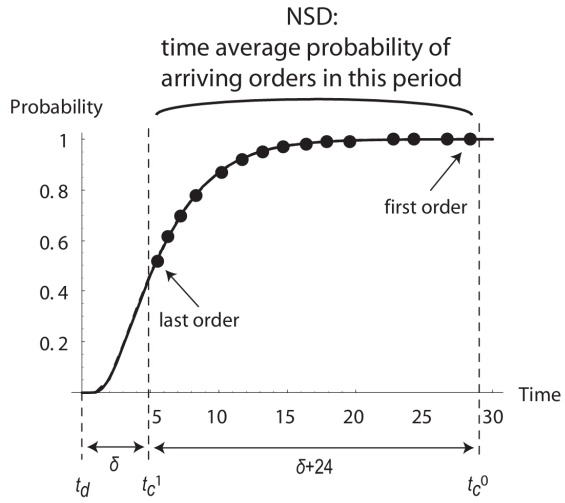

Figure 2. The definition of NSD is grounded in the steady-state sojourn time distribution.

Formally, if $P(t)$ represents the probability of an event occurring at time t, the time-averaged probability is expressed as

$$p = \frac{1}{t}\int_0^t P(t)dt. \quad (1)$$

Proposition 1. *For an order fulfillment system with a deadline t_d and a cutoff time t_c,*

$$NSD = \frac{1}{24}\int_{\delta}^{\delta+24} P[T < t]dt, \tag{2}$$

where $0 < t < 24$ and $\delta = t_d - t_c$.

Proof. Let $\mathrm{P(t)}$ represent the probability that a job arriving at time t is completed before the deadline; this can be expressed as $\mathrm{P(t)} = \mathrm{P[T} < \delta + 24 - t]$ and by the definition of time average probability,

$$\mathrm{NSD} = \frac{1}{24}\int_{\delta}^{\delta+24} \mathrm{P[T} < \delta + 24 - t]dt, \tag{3}$$

where t is elapsed time since the most recent cutoff time $\mathrm{t_c}$. □

Let $\mathrm{u} = \delta + 24 - \mathrm{t}$ and differentiate both sides with respect to t. We obtain $\mathrm{du/dt} = -1$ or $\mathrm{dt} = -\mathrm{du}$.

$$\begin{aligned} \mathrm{NSD} &= \tfrac{1}{24}\textstyle\int_{\delta}^{\delta+24} \mathrm{P[T} < \delta + 24 - t]dt \\ &= \tfrac{1}{24}\textstyle\int_{\delta+24}^{\delta} \mathrm{P[T} < u](-du) \\ &= \tfrac{1}{24}\textstyle\int_{\delta}^{\delta+24} \mathrm{P[T} < u](du) \end{aligned} \tag{4}$$

Given the desired baseline NSD and assuming no worker allocation, δ is determined through a simple search using the NSD equation. Once δ is computed, the cutoff time is established as: $\mathrm{t_c} = \mathrm{t_d} - \delta$.

3.2. State-Dependent Sojourn Time Distribution

State-dependent sojourn time distributions offer dynamic insights into system performance by characterizing the current state, which includes the number of active servers, queued orders, and server processing rates.

Gue et al. [26] introduced an approximation model for sojourn time distributions in multi-server queuing systems. This model accommodates general distributions of interarrival and service times by leveraging phase-type distributions and Markov processes to represent periods of complete server utilization.

In an order fulfillment system, an order in the queue (denoted as the kth) waits through k+1 "sub-waiting times" before entering service and then leaves the system after receiving service. These "sub-waiting times" and the service time are approximated using phase-type distributions, represented as (α_k, T_k) for sub-waiting times and (β, W) for service times. The initial probability vector and the infinitesimal generator of the system's sojourn time distribution, denoted by (γ, Q), are constructed using the convolution property of phase-type distributions. The matrix Q is structured as follows:

$$\mathrm{Q} = \begin{bmatrix} \mathrm{T} & \mathrm{T^0}\alpha_2 & 0 & \cdots & 0 & 0 \\ 0 & \mathrm{T} & \mathrm{T}\alpha_3 & & 0 & 0 \\ & \vdots & & \ddots & \vdots & \\ 0 & 0 & 0 & & \mathrm{T} & \mathrm{T^0}\alpha_{k+1} \\ 0 & 0 & 0 & \cdots & 0 & \mathrm{W} \end{bmatrix}, \tag{5}$$

Based on (γ, Q), the cumulative distribution function (CDF) and the probability density function (PDF) of the sojourn time are defined as

$$\mathrm{F(t)} = \mathrm{P(T} \le \mathrm{t}) = 1 - \gamma \mathrm{e^{Qt}e}, \ (\mathrm{t} \ge 0), \tag{6}$$

$$\mathrm{f(t)} = \gamma \mathrm{e^{Qt}Q^0} = \gamma \mathrm{e^{Qt}}(-\mathrm{Q})\mathrm{e}, (\mathrm{t} \ge 0), \tag{7}$$

where e is a column vector of ones. Refer to [26] for detailed explanations. For our purposes, it is sufficient to understand that the remaining sojourn time distribution for any order in the system can be computed dynamically based on the system's current state.

3.3. The Probability of Success (p_s)

In the introduction, we highlighted that a simple worker allocation policy—moving a fixed number of workers at a set time daily without considering the system state—is not effective. Instead, a more impactful approach is to determine the number of workers to switch based on the current system state. To facilitate this, we introduce the concept of the probability of success (p_s) as a decision-making tool for worker allocation.

The probability of success (p_s) is defined as the likelihood that an arriving order will be completed in time to make it onto the next departing truck. The value of p_s is influenced by several factors, including the current system state, the number of orders ahead in the queue, the number of servers available, the service time distribution, and the remaining time before the deadline. Once an order enters service, the number of orders ahead is no longer relevant. The p_s for an order in the system can be calculated using the state-dependent sojourn time distribution model, as described by [26]. Let T be a continuous random variable representing the sojourn time of an order. Using this model, p_s provides a dynamic measure of the system's ability to meet order deadlines, enabling smarter and more responsive worker allocation policies.

Definition 1. *Probability of success*

$$p_s = \begin{cases} P[S < t_r] \; if \; an \; order \; is \; in \; the \; queue \\ P[S < t_l \,|\, S > t_e] \;\; if \; an \; order \; is \; in \; the \; service \end{cases} \tag{8}$$

where t_r represents the remaining time until the deadline, t_l is the duration of time between an order entering service and the deadline, and t_e denotes the elapsed time since the order began service.

The p_s for an order currently in service can be calculated using the sojourn time CDF as follows:

$$\begin{aligned} P[S < t_l | S > t_e] &= 1 - p[S - t_l | S > t_e] \\ &= 1 - \frac{\int_{t_l}^{\infty} f_S(u)du}{\int_{t_e}^{\infty} f_S(u)du} \\ &= \frac{\int_{t_e}^{\infty} f_S(u)du - \int_{t_l}^{\infty} f_S(u)du}{\int_{t_e}^{\infty} f_S(u)du} \\ &= \frac{\int_{t_e}^{t_l} f_S(u)du}{\int_{t_e}^{\infty} f_S(u)du} \\ &= \frac{F(t_l) - F(t_e)}{1 - F(t_e)}. \end{aligned} \tag{9}$$

For illustration, we monitor p_s for a specific order within a simulated system.

Example 1. *Consider a single-stage queue with 30 identical servers, where the processing times have a mean of 5 h and a squared coefficient of variation of 0.8. An order arrives at 10:00 to find 19 orders ahead in the queue. The last truck leaves the system at 17:00. What is the probability of success for this order, and how does it vary over time?*

To demonstrate how p_s changes over time, we run a simulation model, calculating p_s and the mean sojourn time $E(S)$ at 15 min intervals based on the recalculated state-dependent sojourn time distribution. For instance, to determine p_s for the order at 10:00 (with a remaining time t_r = 7 h), we generate the phase-type representation of the system

(γ, Q) using the approximation model proposed by [26], which reflects the current system state. The value of p_s is then derived using Definition 1.

$$p_s = P(S \leq t_r = 7) = 0.41.$$

Figure 3 illustrates how p_s for the order changes over time. In this scenario, the order remains in the system at the deadline (t_d = 17:00), resulting in $p_s = 0$. The value of p_s fluctuates based on the system's condition. However, once an order enters service, its p_s can only decrease (as defined in Definition 1). This is because once a worker begins processing the order, no additional workers can assist in reducing its processing time, as the model does not account for worker collaboration on a single order.

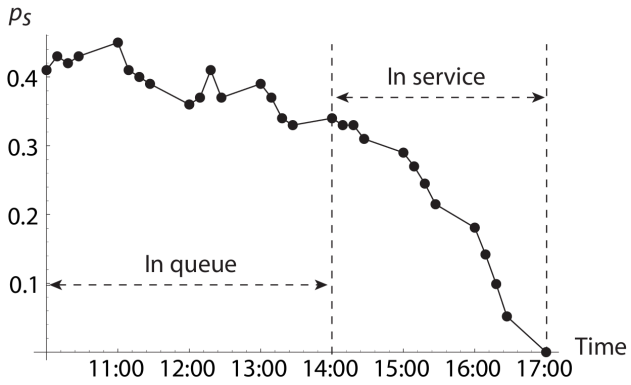

Figure 3. The change in p_s for an order while in the system can be described as follows: p_s fluctuates while the order is in the queue, increasing or decreasing over time as the system state evolves. Factors such as the number of orders ahead, the number of servers, and the service time distribution influence these changes. However, once the order enters service, p_s consistently decreases, as it becomes a conditional probability dependent on the elapsed time t_e.

Our worker allocation policies are grounded in the observation that, as time advances, the probability of success for orders far from completion diminishes. Consequently, it may be more effective to focus effort on orders that are on the cusp of meeting the next truck's deadline. We formalize this concept in the following sections.

4. Worker Allocation Policies

This section presents dynamic worker allocation strategies designed to enhance performance in order fulfillment systems, measured using the NSD metric. We propose two main approaches: the flushing policy and the cascade policy. Flushing policies are further divided into single- and multi-flush variants based on the frequency of worker reallocations.

The single-flush policy reallocates workers to the shipping area at a specific time, prioritizing nearly completed orders to ensure their timely dispatch before the truck departs (Figure 4). This strategy relies on monitoring the state of the shipping area and the sojourn time distributions of orders. However, it offers only one chance daily to resolve bottlenecks in the shipping area. The multi-flush policy addresses this limitation by redistributing workers multiple times throughout the day, based on periodic evaluations of the shipping area. Aside from the increased frequency of reallocations, the multi-flush approach operates similarly to the single-flush policy.

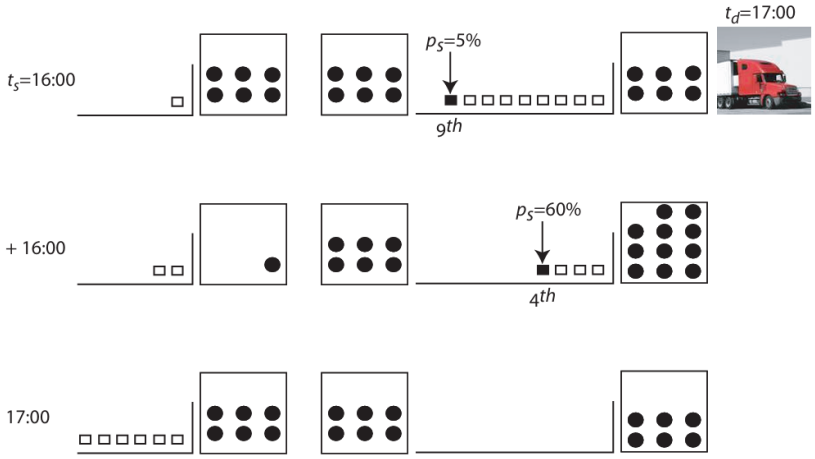

Figure 4. Single-flush policy.

The cascade policy adopts a sequential worker reallocation approach, transferring workers first from picking to packing and later from packing to shipping. This method considers the status of both the picking and shipping areas at designated intervals. Figure 5 illustrates the concept: for example, at a set time each day, workers are moved from picking to packing. Once the backlog in the packing queue is addressed, workers are further reassigned from packing to shipping. The primary aim of this policy is to enhance the probability of success for orders at various stages within the system in a short timeframe.

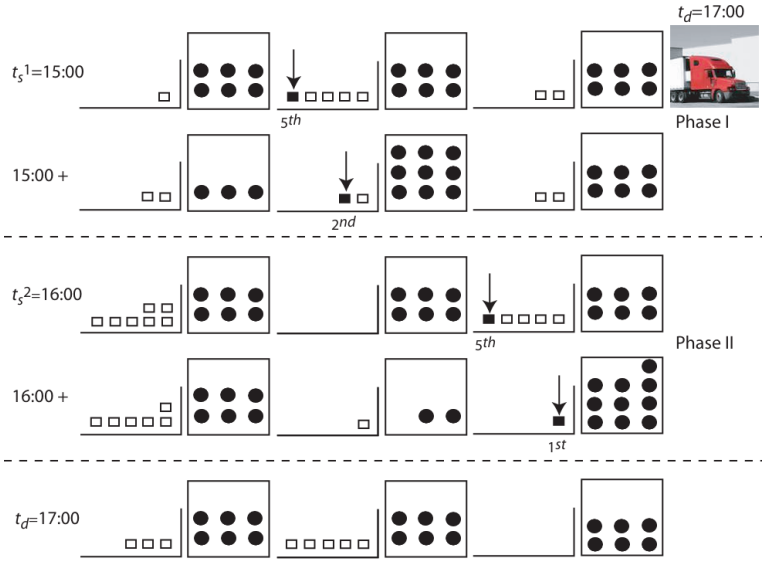

Figure 5. Cascade policy.

5. Experiments

Evaluating the "best" policy across all conceivable scenarios would demand an exhaustive and impractical number of tests. Instead, we designed experiments using 12 specific systems, deliberately constructed to challenge our model across critical parameters. These systems mimic three-stage serial lines, typical of the picking–packing–shipping workflow in most order fulfillment centers. They differ in worker count, average sojourn time, variability in processing times (SCV), and utilization rates. A summary of these configurations is presented in Table 1.

Table 1. The feature of the 12 systems.

Category	High ρ High SCV	High ρ Low SCV	Low ρ High SCV	Low ρ Low SCV
Small system with short $E[S]$	System 1	System 2	System 3	System 4
Large system with short $E[S]$	System 5	System 6	System 7	System 8
Small system with long $E[S]$	System 9	System 10	System 11	System 12

We identified three primary candidate systems based on their size and mean sojourn time, and then further categorized each into four variants by considering two levels of the squared coefficient of variation (SCV = 0.5, 0.9) and utilization (ρ = 0.85, 0.95). Since SCV values below 1 are more typical in practice, our analysis focuses exclusively on this range. The three selected serial lines include a small system with 31 workers and a short mean sojourn time of 6.52 h; a large system with 126 workers and a short mean sojourn time of 7.31 h; and another small system with 44 workers but a long mean sojourn time of 19.76 h. Details for representative systems 1, 5, and 9 are summarized in Table 2.

Table 2. System information for systems 1, 5, and 9.

System 1	***E*[*T*]**	**SCV**	ρ	**Number of workers**
Interarrival	0.117	0.75		
Picking	1.07	0.9	0.91	10
Packing	1.3	0.9	0.93	12
Shipping	1.0	0.9	0.95	9
System 5	***E*[*T*]**	**SCV**	ρ	**Number of workers**
Interarrival	0.053	0.75		
Picking	2.7	0.9	0.92	56
Packing	1.5	0.9	0.95	30
Shipping	2.0	0.9	0.95	40
System 9	***E*[*T*]**	**SCV**	ρ	**Number of workers**
Interarrival	0.23	0.75		
Picking	3.3	0.9	0.96	15
Packing	4.3	0.9	0.93	20
Shipping	2.0	0.9	0.97	9

E[T]: mean processing time.

To effectively implement a worker allocation policy, two key questions must be addressed: how many workers should be reallocated, and when? Allocating workers too early or in excessive numbers may disrupt system balance, while reallocating them too late or in insufficient numbers might fail to improve the NSD.

The exact number of workers to reallocate depends on the target probability of success p_s for a specific order and the switching time t_s. To determine the optimal t_s and p_s for each policy, we tested 12 systems. For instance, as shown in Figure 4, the probability of success p_s for the last order in the shipping queue is initially 5%, considering 8 orders ahead, 6 workers, and a remaining time t_r = 1 h ($t_d - t_s$). To achieve a target p_s of 60%, 5 additional workers are assigned to shipping. This adjustment changes the system state

to 4 orders ahead and 11 workers. The required number of workers is determined through trial and error using an approximation model for state-dependent sojourn time distributions in multi-server queuing systems ([26]).

$$\text{Target } p_s = 60\% = P(T \leq t_r = 1) = 1 - \gamma e^{Kt_r} e,$$

where (γ, K) depends on the system state.

We evaluate six target probability levels p_s (10%, 30%, 50%, 60%, 70%, 90%). Second, we use switching time t_s from early in the morning to close to the deadline. We do not include a switching time that makes the remaining time t_r less than the mean processing time of the last workstation because this situation limits the effect of the policy (more on this below). Additionally, we test switching times t_s ranging from early morning to just before the deadline. However, switching times that result in a remaining time t_r shorter than the mean processing time of the final workstation are excluded, as such scenarios would significantly reduce the policy's effectiveness (explained in more detail below).

We conducted simulations using Arena 12.0, with each scenario simulated 100 times over a 50-day period. To streamline the simulation process, we precalculated the exact number of workers required to achieve a given target probability (p_s) for all possible combinations of target p_s and switching time (t_s) for each system. During the simulation, worker reallocations were determined based on the current queue size in the shipping area and the precalculated information. For example, in System 1, if the target p_s is 60% and the switching time (t_s) is 16:00, and there are 4 orders in the shipping queue at that time, we reallocate 4 workers from picking to shipping as per the precomputed requirements. The simulation model relies on two key assumptions for simplicity:

1. Worker transition times between workstations are not considered, as these are highly application-specific and vary significantly in practice.
2. The picking process is treated as non-batch, which is not universally accurate but is reasonable for certain order fulfillment systems, particularly part-to-picker systems.

5.1. Single Flush

The truck departure time for all experiments is set at 17:00. To determine the optimal switching time (t_s) and target probability (p_s) for the single-flush policy, the following conditions are evaluated:

- Target p_s: {10%, 30%, 50%, 60%, 70%, 90%};
- Target p_s: {10%, 30%, 50%, 60%, 70%, 90%};
- Switching time t_s
- Systems 1~4: {16:00, 15:00, 14:00, 13:00, 07:00};
- Systems 5~8: {15:00, 14:00, 13:00, 12:00, 07:00};
- Systems 9~12: {15:00, 14:00, 13:00, 12:00, 07:00, 02:00}.

The candidate switching times range from early in the morning (02:00 or 07:00) to close to the deadline (16:00 or 15:00). However, switching times later than 16:00 for Systems 1~4 or 15:00 for Systems 5~12 are excluded because the mean processing time of the shipping area ($E[T]_3$) is typically 1~2 h. When the remaining time (t_r) is less than $E[T_3]$, the probability of success (p_s) at that switching time becomes too low to meaningfully impact NSD. The Algorithm 1 for the Single Flush as follows:

Algorithm 1: Single Flush (SF)

- **Step 1:** Determine the cutoff time (t_c) using the steady-state sojourn time distribution based on the baseline NSD.
- **Step 2:** At a given switching time (t_s), check the number of orders queued in the shipping area.
- **Step 3:** Calculate the probability of success (p_s) for the last order in the shipping queue using the state-dependent sojourn time distribution.
- **Step 4:** If p_s is below the specified target value, calculate how many workers need to be reallocated to achieve the target p_s, and transfer the required number of workers from picking to shipping. If p_s meets or exceeds the target, no workers are moved.
- **Step 5:** Return all reallocated workers to their original stations when the clock reaches the deadline (t_d).

Initially, the cutoff time (t_c) is determined using a steady-state sojourn time distribution model. For example, under baseline conditions (without worker reallocation) in System 1, the NSD is set to 77%. This serves as the reference point for calculating t_c and evaluating worker allocation adjustments.

$$\text{Baseline NSD} = 77\% = \frac{1}{24}\int_{\delta}^{24+\delta} \mathrm{P}[\mathrm{T} < t]\mathrm{dt}$$

$$\delta \approx 1\ \mathrm{h}$$

$$t_c = 16:00.$$

Next, we evaluate the single-flush policy across various combinations of switching times (t_s) and target probabilities (p_s), resulting in a total of 384 scenarios for Systems 1~9. Table 3 presents the results for 30 scenarios specific to System 1. The comparison between the single-flush policy and the fixed worker model is conducted using three key metrics: mean sojourn time (E[S]]), expected number of switching workers (E[SW]), and NSD. The results show that the NSD for the single-flush policy consistently outperforms the fixed model across all combinations of t_s and p_s. NSD improvements range from 0.25% to 4.01%. Additionally, the mean sojourn time (E[S]) is reduced by 0.07 to 0.84 h compared with the fixed worker model, further demonstrating the efficiency of the single-flush policy.

Table 3. Outcomes of the single-flush policy for system 1.

Category	**Fixed**	t_s **= 16:00**					
		p_s **= 10%**	p_s **= 30%**	p_s **= 50%**	p_s **= 60%**	p_s **= 70%**	p_s **= 90%**
$E[S]$	6.53	5.76	5.70	5.69	5.71	5.71	5.71
$E[SW]$	0	2.27	3.60	4.50	5.12	5.12	5.12
NSD	77.25	80.56	81.10	81.05	81.26	81.26	81.26
Category	**Fixed**	t_s **= 15:00**					
		p_s **= 10%**	p_s **= 30%**	p_s **= 50%**	p_s **= 60%**	p_s **= 70%**	p_s **= 90%**
$E[S]$	6.53	5.94	5.88	5.83	5.85	5.84	6.10
$E[SW]$	0	0.81	1.22	1.66	1.99	2.17	5.24
NSD	77.25	80.10	80.41	80.89	80.72	81.03	80.77

Table 3. *Cont.*

Category	Fixed	t_s = 14:00					
		p_s = 10%	p_s = 30%	p_s = 50%	p_s = 60%	p_s = 70%	p_s = 90%
$E[S]$	6.53	6.08	6.03	6.01	5.99	6.00	6.06
$E[SW]$	0	0.34	0.52	0.65	0.79	1.02	1.83
NSD	77.25	79.35	79.51	79.88	79.90	80.24	80.45
Category	**Fixed**	t_s = 13:00					
		p_s = 10%	p_s = 30%	p_s = 50%	p_s = 60%	p_s = 70%	p_s = 90%
$E[S]$	6.53	6.10	6.15	6.10	6.11	6.08	6.05
$E[SW]$	0	0.32	0.23	0.29	0.36	0.44	0.75
NSD	77.25	79.08	78.91	79.42	79.32	79.42	79.87
Category	**Fixed**	t_s = 07:00					
		p_s = 10%	p_s = 30%	p_s = 50%	p_s = 60%	p_s = 70%	p_s = 90%
$E[S]$	6.53	6.37	6.46	6.46	6.46	6.46	6.43
$E[SW]$	0	0.14	0	0	0	0	0
NSD	77.25	78.24	77.40	77.53	77.55	77.48	77.54

In Table 3, it is evident that the target p_s = 60, 70, 90% for t_s = 16:00 produce identical results. This phenomenon is explained in Figure 6. When the remaining time (t_r) is just 1 h, the maximum achievable p_s for any order in the shipping queue under a worker allocation policy is approximately 60%. This limitation arises because, when t_r equals the mean processing time of the shipping workstation ($E[T_3]$), it is impossible to increase the p_s of the last order beyond about 60%, regardless of how many additional workers are allocated. Here is why: Suppose there are m orders in the shipping queue at t_s, and the time remaining equals $E[T_3]$. Moving more than m workers from picking to shipping would be ineffective, as any workers beyond m would have no orders to process. By reallocating m workers, the last order in the shipping queue would enter service with exactly $E[T_3]$ time remaining. Its p_s would then increase to $P[T_3 < E[T_3]]$. For a symmetric distribution, this probability would be 50%, but since the shipping processing time (T_3) is modeled as an Erlang (k, μ) distribution (with SCV<1), the probability $P[T_3 < E[T_3]]$ is approximately 0.6 under the tested parameters. Thus, target p_s values exceeding 60% are unachievable when reallocating workers with only $E[T_3]$ time remaining. This finding highlights the importance of considering the remaining time and system characteristics when setting target probabilities in worker allocation policies.

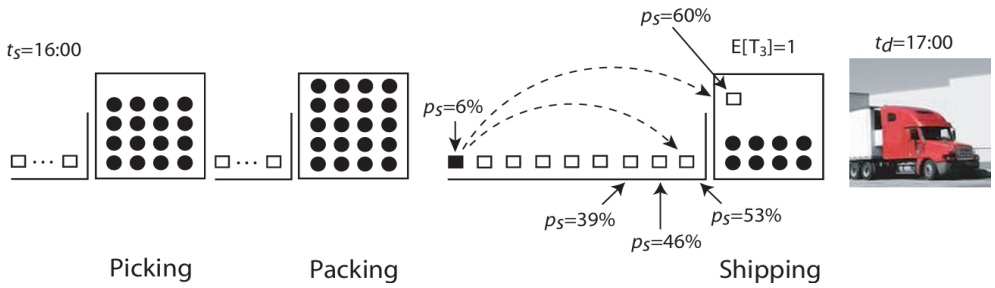

Figure 6. The maximum p_s an order can achieve through worker allocation is as follows: Adding 6 workers to the shipping area increases the p_ss of the last order in the shipping queue to 39%. Adding 9 workers raises p_s to 60%. Allocating more than 9 workers results in idle workers without further increasing p_s.

As shown in Table 3, the NSD for the same target p_s decreases when the switching time t_s is set earlier. Table 4, which summarizes data for all 12 systems with p_s = 60%, confirms

this pattern. Switching too early is ineffective because workers who are reassigned often remain idle in shipping once the conditions necessitating their movement have resolved. Furthermore, at earlier times, very few workers switch since the p_s values of orders in the shipping queue are generally still quite high.

Table 4. Change in NSD according to switching times t_s with p_s = 60%.

t_s \ System	1	2	3	4	5	6	7	8	9	10	11	12
15:00 (16:00)	81.3	82.7	87.5	88.0	78.0	78.6	81.9	82.0	80.4	77.6	70.0	70.7
14:00 (15:00)	80.7	82.1	87.3	87.8	77.7	78.2	81.9	82.0	79.7	76.3	69.8	70.5
13:00 (14:00)	79.9	81.7	87.1	87.7	77.3	77.9	81.8	81.9	79.3	76.1	69.5	70.3
12:00 (13:00)	79.3	81.1	87.0	87.7	76.9	77.8	81.8	81.9	78.1	74.9	69.3	70.1
07:00	77.6	80.2	86.9	87.6	76.7	77.7	81.8	81.9	75.5	72.1	68.8	69.9
02:00	-	-	-	-	-	-	-	-	73.6	70.3	68.8	69.9

Unlike the switching time t_s, there appears to be no significant difference in NSD among policies with varying target p_s values at the same t_s (Table 3). To determine the optimal target p_s, we conducted a series of *t*-tests comparing scenarios with different p_s values, each based on 100 simulation runs ([27]). For an early switching time of 07:00, there is no statistically significant difference among p_s values, as very few workers switch (average $E[SW] \leq 0.14$ workers), making it challenging to improve NSD beyond the fixed worker model. However, for later switching times, higher p_s values yield better results. This trend is evident in Figure 7, which highlights the best scenarios across all 12 systems based on *t*-test results. The figure excludes target p_s values that exceed the maximum achievable p_s, leaving those areas blank.

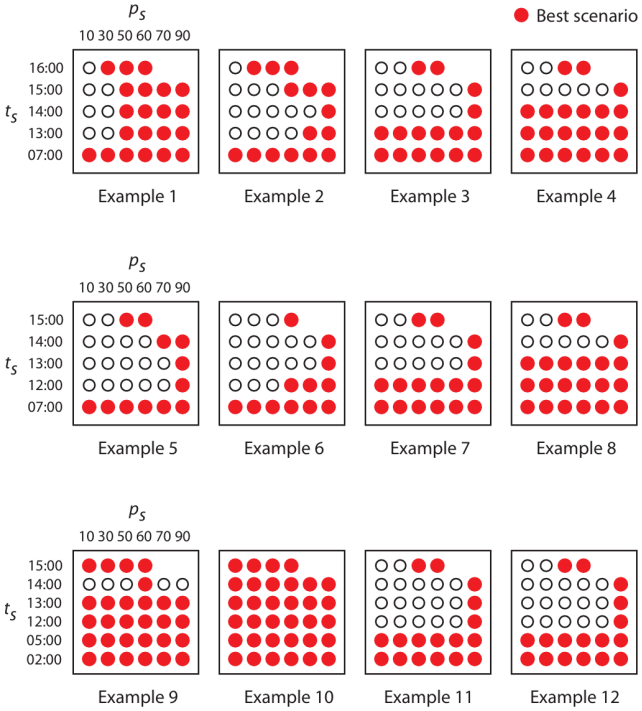

Figure 7. Best scenarios for all 12 Systems: open circles indicate good scenarios; gray circles denote best scenarios; empty space represents cases where the NSD is identical to that of the next lower p_s.

In the graphics presented in Figure 7, we observe a distinct pattern (illustrated in Figure 8): when the switching time is near the deadline, nearly any p_s value proves effective.

This is because the situation where the p_s value of orders in the shipping queue falls below the target p_s at the switching time is more frequent. Conversely, when the switching time is very early, all p_s values appear to be "best" simply because none are truly effective. The early switching time prevents any significant impact on NSD, regardless of the selected p_s value.

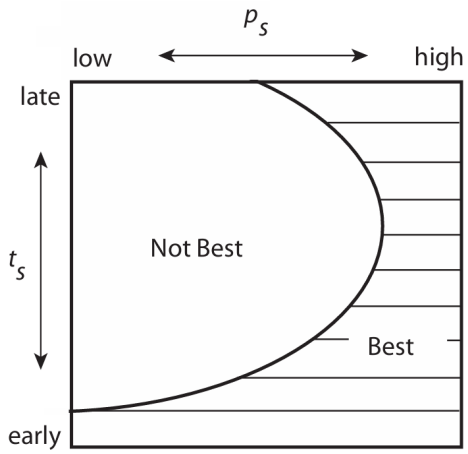

Figure 8. An observed trend in the scenarios of Figure 7.

In summary, the single-flush policy proves highly effective in significantly increasing NSD while reducing the mean sojourn time E[S]. A later switching time t_s combined with a higher target p_s consistently yields the best results. Specifically, the optimal condition for the single flush is achieved with a target p_s = 60% at a switching time $t_s = t_d - E[T_3]$. Notably, this setup involves moving a number of workers equal to the number of orders in the shipping queue at the switching time. This highlights a straightforward yet powerful approach for implementing an effective policy.

5.2. Multi-Flush

We set the switching period t_p equal to the mean processing time of the shipping area $E[T_3]$, as our analysis of single-flush policies demonstrated that a later switching time t_s is more effective than an earlier one. Consequently, the number of workers allocated during each switching period is determined dynamically based on the system state, particularly when the remaining time $t_r = t_p = E[T_3]$. To identify the optimal target p_s for the multi-flush policy, we evaluate six target p_s values (10%, 30%, 50%, 60%, 70%, 90%), consistent with the single-flush experiment. The number of switching events ranges from two to five. The Algorithm 2 for the Multi Flush proceeds as follows:

Algorithm 2: Multi-Flush (MF)

- **Step 1:** Determine the cutoff time (t_c) using the steady-state sojourn time distribution based on the baseline NSD.
- **Step 2**: At every switching period t_p, check the number of orders in the shipping area.
- **Step 3**: Calculate p_s for the last order in the shipping queue using the state-dependent sojourn time distribution model, with $t_r = t_p$.
- **Step 4**: If p_s is below the target p_s calculate the number of workers needed to reach the target p_s and transfer these workers from the picking area to the shipping area. If p_s is not below the target, do not move workers.
- **Step 5**: If the clock time has not yet reached the deadline but has reached the next switching period, return workers to the picking area and repeat from Step 2. Otherwise, proceed to Step 6.
- **Step 6**: When the clock time reaches the deadline t_d, return all workers to their original positions.

We compared the system performance of the multi-flush policy with the fixed model, as presented in Table 5, which includes test results for System 1. The multi-flush policy demonstrates significant improvements in performance, with NSD increasing by 4.16% to 5.58% and E[S] decreasing by 0.88 to 0.98 h compared with the fixed model, regardless of the number of switching events. However, increasing the number of switching events involves reallocating more workers, which suggests that fewer switching events may be preferable in practical applications. Additionally, it is noteworthy that target p_s values of 60%, 70%, and 90% yield identical results for every switching t_s. This mirrors the pattern observed with the single-flush policy, where higher p_s values produced similar outcomes beyond a certain threshold.

Table 5. Multi-flush results for System 1.

Category	**Fixed**	**2Times (t_s = 16:00, 15:00)**					
		p_s = 10%	**p_s = 30%**	**p_s = 50%**	**p_s = 60%**	**p_s = 70%**	**p_s = 90%**
$E[S]$	6.53	5.73	5.68	5.65	5.68	5.68	5.68
$E[SW]$	0	2.73	3.89	5.40	6.20	6.20	6.20
NSD	77.25	81.42	82.01	82.26	82.40	82.40	82.40
Category	**Fixed**	**3Times (t_s = 16:00, 15:00, 14:00)**					
		p_s = 10%	**p_s = 30%**	**p_s = 50%**	**p_s = 60%**	**p_s = 70%**	**p_s = 90%**
$E[S]$	6.53	5.69	5.65	5.65	5.64	5.64	5.64
$E[SW]$	0	3.06	4.32	6.16	7.04	7.04	7.04
NSD	77.25	81.61	82.30	82.73	82.77	82.87	82.87
Category	**Fixed**	**4Times (t_s = 16:00, 15:00, 14:00, 13:00)**					
		p_s = 10%	**p_s = 30%**	**p_s = 50%**	**p_s = 60%**	**p_s = 70%**	**p_s = 90%**
$E[S]$	6.53	5.66	5.62	5.62	5.64	5.64	5.64
$E[SW]$	0	3.30	4.70	6.79	7.89	7.89	7.89
NSD	77.25	81.99	82.83	82.75	82.65	82.65	82.65
Category	**Fixed**	**5Times (t_s = 16:00, 15:00, 14:00, 13:00, 12:00)**					
		p_s = 10%	**p_s = 30%**	**p_s = 50%**	**p_s = 60%**	**p_s = 70%**	**p_s = 90%**
$E[S]$	6.53	6.10	6.15	6.10	6.11	6.08	6.05
$E[SW]$	0	3.52	5.04	7.34	8.68	8.68	8.68
NSD	77.25	81.83	82.30	82.45	82.66	82.66	82.66
Category	**Fixed**	**6Times (t_s = 16:00, 15:00, 14:00, 13:00, 12:00, 11:00)**					
		p_s = 10%	**p_s = 30%**	**p_s = 50%**	**p_s = 60%**	**p_s = 70%**	**p_s = 90%**
$E[S]$	6.53	5.62	5.56	5.57	5.63	5.63	5.63
$E[SW]$	0	3.70	5.33	7.94	9.46	9.46	9.46
NSD	77.25	82.28	82.35	82.46	82.49	82.49	82.49

To determine the optimal target p_s among 10%, 30%, 50%, and 60% for the same switching time t_s, we conducted a series of t-tests comparing these target values. The results consistently indicated that the highest target, p_s = 60%, was the best-performing scenario. Based on this analysis, we conclude that the optimal configuration for the multi-flush policy, considering system stability and simplicity, is a target p_s = 60%, $t_r = E[T_3]$, and two switching events. This setup strikes a balance between effectiveness and operational feasibility.

5.3. Cascade Policy

The cascade policy optimizes resource allocation by dynamically reallocating workers based on system conditions at two sequential switching times. The target p_s is set to 60%, with the switching times determined by the mean processing times of the packing and shipping areas. At the first switching time, the state of the packing area is assessed, and

the required number of workers is moved from the picking area to achieve the target p_s, based on the remaining time in the packing process. Similarly, at the second switching time, the state of the shipping area is evaluated, and workers are reallocated from the packing area to the shipping area to maintain the target p_s as the system progresses. Throughout the process, the system state is monitored to ensure the target p_s is sustained. Worker allocations are adjusted dynamically as necessary. At the end of the operational period, all workers return to their original positions. This policy ensures efficient and balanced resource utilization while maintaining stability across the system. The Algorithm 3 for the Cascade as follows:

Algorithm 3: Cascade (C)

- **Step 1**: Calculate the cutoff time t_c using the steady-state sojourn time distribution model, referencing the baseline NSD.
- **Step 2**: At ${t_s}^1 = t_d - (E[T_2] - E[T_3])$, evaluate the number of orders in front of the packing area.
- **Step 3**: Determine the p_s value of the last order in the packing queue using the state-dependent sojourn time distribution with $t_r = E[T_2]$.
- **Step 4**: If p_s is below the target p_s calculate the number of workers needed to meet the target p_s and reallocate them from the picking area to the packing area. Otherwise, do not move workers.
- **Step 5**: When the clock reaches ${t_s}^2 = t_d - E[T_3]$, return workers to their original positions and assess the number of orders in front of the shipping area.
- **Step 6**: Determine the p_s value of the last order in the shipping queue using the state-dependent sojourn time distribution with $t_r = E[T_3]$.
- **Step 7**: Otherwise, do not move workers. If p_s is below the target p_s calculate the number of workers needed to meet the target p_s and reallocate them from the packing area to the shipping area. Otherwise, do not move workers.
- **Step 8**: When the clock reaches the final deadline t_d, return all workers to their original positions.

Table 6 compares the system performance of three policies: the cascade policy, a single flush with a target p_s = 60% and $t_s = t_d - E[T_3]$, and a multi-flush with a target p_s = 60%, two switching events, and a switching period of $t_r = E[T_3]$, across all 12 systems. Overall, the performance differences among the three policies are minimal. However, the cascade policy underperforms in Systems 9 and 10 due to the extended switching duration of 6.3 h ($E[T_2] + E[T_3]$), which is significantly longer than the single- and multi-flush policies' switching duration ($E[T_3]$ = 2 h). This prolonged absence from the picking station results in a very high average queue level at picking (44.7 and 50.5 orders). Additionally, Systems 9 and 10 exhibit high utilization and high variability, which further contribute to large queues in front of each workstation.

Table 7 also reveals that dynamic worker allocation policies are more effective in systems with higher utilization (ρ) and greater variability (SCV), as these systems typically have longer queues.

Table 6. Results of the Cascade Policy and Comparison Across Three Policies.

Category		System 1				System 2				System 3				System 4			
		F	**SF**	**MF**	**C**	**F**	**SF**	**MF**	**C**	**F**	**SF**	**MF**	**C**	**F**	**SF**	**MF**	**C**
$E[S]$		6.5	5.7	5.7	5.6	5.8	5.2	5.2	5.2	4.2	4.1	4.1	4.0	4.0	3.9	3.9	3.9
$E[SW]$		0	5.1	6.2	9.5	0	4.7	5.5	8.4	0	2.3	2.8	5.0	0	1.8	2.4	4.4
NSD		77.3	81.3	82.4	82.5	80.0	82.9	84.1	83.9	86.9	87.6	88.0	88.0	87.6	88.1	88.4	88.4
Average orders in queue	Picking area	5.8	7.2	8.2	8.0	5.0	6.1	7.2	6.8	1.7	1.8	1.9	1.9	1.4	1.4	1.5	1.5
	Packing area	7.4	7.8	6.0	7.8	5.4	5.6	4.6	5.6	1.9	1.9	1.8	1.9	1.3	1.4	1.3	1.4
	Shipping area	12.6	5.4	5.1	3.8	9.6	4.2	4.0	3.1	2.8	2.1	2.1	1.7	2.0	1.6	1.6	1.3
Category		**System 5**				**System 6**				**System 7**				**System 8**			
		F	**SF**	**MF**	**C**	**F**	**SF**	**MF**	**C**	**F**	**SF**	**MF**	**C**	**F**	**SF**	**MF**	**C**
$E[S]$		7.6	7.3	7.2	7.2	7.4	7.1	7.1	7.0	6.3	6.3	6.3	6.3	6.3	6.3	6.3	6.3
$E[SW]$		0	7.4	8.4	13.2	0	6.1	7.1	11.7	0	1.1	1.5	3.5	1	0.8	1.3	3.0
NSD		76.7	78.5	78.7	78.9	77.7	78.9	79.3	81.8	82.0	82.0	82.0	81.9	82.1	82.1	82.1	82.0
Average orders in queue	Picking area	3.8	4.4	4.4	3.9	3.2	3.8	3.8	3.4	0.3	0.3	0.3	0.3	0.2	0.3	0.3	0.3
	Packing area	12.3	12.2	12.2	11.0	10.3	10.2	10.2	9.7	1.3	1.4	1.4	1.4	1.1	1.1	1.1	1.1
	Shipping area	11.9	2.6	2.6	4.9	8.7	2.3	2.3	4.1	1.0	0.6	0.6	0.9	0.8	0.5	0.5	0.7
Category		**System 9**				**System 10**				**System 11**				**System 12**			
		F	**SF**	**MF**	**C**	**F**	**SF**	**MF**	**C**	**F**	**SF**	**MF**	**C**	**F**	**SF**	**MF**	**C**
$E[S]$		19.5	16.5	16.6	21.7	17.8	15.5	15.6	22.7	11.1	10.8	10.8	10.9	10.8	10.6	10.6	10.7
$E[SW]$		0	5.5	6.0	11.0	0	4.9	5.3	9.9	0	2.3	2.4	4.6	0	1.9	2.4	3.8
NSD		70.3	80.4	81.1	75.4	68.6	77.2	77.5	74.1	68.8	70.3	70.7	70.1	69.9	71.0	71.6	70.9
Average orders in queue	Picking area	13.9	18.4	20.8	44.7	13.0	16.9	20,2	50.5	2.0	2.0	2.1	2.4	1.6	1.6	1.7	2.0
	Packing area	8.7	8.5	8.4	5.0	6.1	5.8	5.7	3.6	1.2	1.3	1.3	1.1	0.9	0.9	0.9	0.8
	Shipping area	20.9	3.6	1.8	3.5	16.8	3.0	1.6	3.0	2.9	1.7	0.9	1.7	2.1	1.4	0.8	1.4

F: fixed model, SF: single flush, MF: multi-flush, C: cascade.

Table 7. System Performance Based on System Characteristics.

	Category		**High ρ High SCV**	**High ρ Low SCV**	**Low ρ High SCV**	**Low ρ Low SCV**
			System 1	**System 2**	**System 3**	**System 4**
Small system with short $E[S]$	Average orders in queue	Picking area	5.8	5.0	1.7	1.4
		Packing area	7.4	5.4	1.9	1.3
		Shipping area	12.6	9.6	2.8	2.0
	Increased in NSD * (%)		4.01	2.94	0.66	0.51
			System 5	System 6	System 7	System 8
Large system with short $E[S]$	Average orders in queue	Picking area	3.8	3.2	0.3	0.2
		Packing area	12.3	10.3	1.3	1.1
		Shipping area	11.9	8.7	1.0	0.8
	Increased in NSD * (%)		1.79	1.25	0.16	0.11
			System 9	System 10	System 11	System 12
Small system with long $E[S]$	Average orders in queue	Picking area	13.9	13.0	2.0	1.6
		Packing area	8.7	6.1	1.2	0.9
		Shipping area	20.9	16.8	2.9	2.1
	Increased in NSD * (%)		10.08	8.63	1.51	1.08

* Fixed model—single flush.

Comparing the three policies, we find that the multi-flush and cascade policies do not offer significant advantages over the single-flush policy, except in Systems 9 and 10. However, both the multi-flush and cascade policies require more workers switching compared with the single-flush policy, potentially making them more challenging to implement. If worker switching is relatively easy to manage, or if even a small performance improvement, such as 1%, is critical for the system, the multi-flush policy could be a viable choice.

6. Conclusions

In this study, we proposed several dynamic worker allocation policies for due-date order fulfillment systems. Our findings demonstrate that dynamic worker allocation can enhance service performance (NSD) while reducing mean sojourn time. This confirms that effective worker reallocation can improve service performance without disrupting system stability or balance.

We introduced single-flush, multi-flush, and cascade policies as potential dynamic worker allocation strategies. Through comprehensive testing across 12 systems, we identified the best policy, switching times, and target probability of success. Our results show that later switching times, with shorter remaining time, have a greater impact on improving NSD compared with earlier times. Furthermore, higher target probabilities of success enhance NSD by increasing opportunities for worker allocation, utilizing more workers when needed. This approach maintains system balance, as the number of allocated workers is determined dynamically based on the system state at each switching time.

Using the optimal switching time and target probability of success, we tested and compared the three policies and found minimal differences in overall system performance among them. While the multi-flush and cascade policies (except for cases with long switching durations or early switching times) perform slightly better than the single flush, we believe the single flush is the most practical and reliable option in terms of system stability and ease of implementation.

Another key finding is that systems with longer waiting times have greater potential for improving NSD. High utilization and variation are significant contributors to system delays, making these factors critical targets for improvement. Systems with high utilization

and variation benefit the most from our dynamic worker allocation policies, suggesting that the methods developed in this research are particularly effective for systems operating under heavy traffic conditions, where the potential for performance gains is greatest.

While the policies we developed rely heavily on state-dependent sojourn time distributions, our experimental results point to a straightforward policy that eliminates the need for complex calculations: (1) Set the switching time to the deadline minus the mean processing time of the shipping area. (2) At this switching time each day, transfer a number of workers from picking to shipping equal to the size of the shipping queue, without exceeding the total number of workers in picking. This simple approach closely mirrors the single-flush policy with a target p_s = 60% and switching time $t_s = t_d - E[T]_3$. As such, we can expect it to deliver performance comparable to the more mathematically driven single-flush policy, making it a practical and effective option.

There are limitations due to several assumptions in this study. First, it is assumed that workers are cross-trained and can perform any task and can move at any time, and there is no time lag to move between workstations. It is realistically limited to configure workers who can perform tasks at all workstations and move between workstations without any time difference. In addition, this study did not consider the cost of having one worker do multiple tasks. Second, it is assumed that the balance of the system will be stable even if the position of the worker is changed, but in a real system, it may be difficult to recover work efficiency due to the preparation time of the workers or equipment and the learning curve. In the future, system performance measurement that considers the mentioned limitations is required.

Author Contributions: Methodology, H.K. and W.K.; formal analysis, H.K. and W.K.; data curation, H.K.; writing—original draft, H.K.; writing—review & editing, W.K. and E.L.; visualization, H.K.; supervision, H.K.; project administration, E.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the 2025 research fund of the Korea Military Academy (Hwarangdae Research Institute). This research was funded by Kookmin University.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Acknowledgments: This study significantly revises and extends the conceptual framework and methodology derived from Hyunho Kim's dissertation, *Modeling Service Performance and Dynamic Worker Allocation Policies for Order Fulfillment Systems (2009)*. The research builds upon the foundational concepts and methodologies presented in the dissertation, incorporating substantial modifications and refinements to enhance the analytical framework and empirical applicability.

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

References

1. Bartholdi, J.J., III; Eisenstein, D.D. A production line that balances itself. *Oper. Res.* **1996**, *44*, 21–34. [CrossRef]
2. Doerr, K.H.; Gue, K.R. A performance metric and goal-setting procedure for deadline-oriented processes. *Prod. Oper. Manag.* **2013**, *22*, 726–738. [CrossRef]
3. Kaplan, R.S.; Norton, D.P. The balanced scorecard: Measures the drive performance. *Harv. Bus. Rev.* **1992**, *70*, 71–79. [PubMed]
4. Osterwalder, A.; Pigneur, Y. *Business Model Generation*; John Wiley and Sons: Hoboken, NJ, USA, 2010.
5. Askin, R.G.; Chen, J. Dynamic task assignment for throughput maximization with worksharing. *Eur. J. Oper. Res.* **2006**, *168*, 853–869. [CrossRef]
6. Bischak, D.P. Performance of a manufacturing module with moving workers. *IIE Trans.* **1996**, *28*, 723–733. [CrossRef]

7. Zavadlav, E.; McClain, J.O.; Thomas, L.J. Self-buffering, self-balancing, self-flushing production lines. *Manag. Sci.* **1996**, *42*, 1151–1164. [CrossRef]
8. Bartholdi, J.J., III; Eisenstein, D.D.; Foley, R.D. Performance of Bucket Brigades When Work is Stochastic. *Oper. Res.* **2001**, *49*, 710–719. [CrossRef]
9. McClain, J.O.; Schultz, K.L.; Thomas, L.J. Management of worksharing systems. *Manuf. Serv. Oper. Manag.* **2000**, *2*, 49–67. [CrossRef]
10. Gel, E.S.; Hopp, W.J.; Van Oyen, M.P. Hierarchical cross-training in work-in-process-constrained systems. *IIE Trans.* **2007**, *39*, 125–143. [CrossRef]
11. Gel, E.S.; Hopp, W.J.; Van Oyen, M.P. Factors affecting opportunity of worksharing as a dynamic line balancing mechanism. *IIE Trans.* **2002**, *34*, 847–863. [CrossRef]
12. Ostolaza, J. The use of dynamic (state-dependent) assemblyline balancing to improve throughput. *J. Manuf. Oper. Manag.* **1990**, *3*, 105–133.
13. Chen, J.; Askin, R.G. Throughput maximization in serial production lines with worksharing. *Int. J. Prod. Econ.* **2006**, *99*, 88–101. [CrossRef]
14. Andradottir, S.; Ayhan, H.; Kirkizlar, E. Flexible servers in tandem lines with setup costs. *Queueing Syst.* **2011**, *70*, 165–186. [CrossRef]
15. Tekin, S.; Andradottir, S.; Down, D.G. Dynamic server allocation for unstable queueing networks with flexible servers. *Queueing Syst.* **2011**, *70*, 45–79. [CrossRef]
16. Isik, T.; Andrad'ottir, S.; Ayhan, H. Optimal control of queueing systems with non-collaborating servers. *Queueing Syst.* **2016**, *84*, 79–110. [CrossRef]
17. Kırkızlar, E.; Andradottir, S.; Ayhan, H. Profit maximization in flexible serial queueing networks. *Queueing Syst.* **2014**, *77*, 427–464. [CrossRef]
18. Ganbold, O.; Kundu, K.; Li, H.; Zhang, W. A Simulation-Based Optimization Method for Warehouse Worker Assignment. *Algorithms* **2020**, *13*, 326. [CrossRef]
19. Liu, M.; Chen, Y.; Wang, T. Collaborative multi-agent systems for dynamic scheduling in high-density environments. *J. Manuf. Syst.* **2022**, *62*, 292–301.
20. Kim, J.; Park, H.; Shin, Y. Bayesian adaptive queuing models for dynamic resource allocation. *J. Oper. Res.* **2020**, *68*, 234–248.
21. Huang, L.; Lee, C. Workforce cross-training and its impact on service quality: Evidence from multi-stage operations. *Int. J. Prod. Econ.* **2023**, *254*, 108644.
22. Lam, T.; Chen, W.; Li, J. Predictive analytics in workforce management: A review of applications. *Eur. J. Oper. Res.* **2021**, *294*, 450–465.
23. Wang, T.; Liu, J.; Zhou, H. Real-time digital twins for optimizing dynamic task allocations. *Comput. Oper. Res.* **2023**, *160*, 105859.
24. Chu, R.; Zhao, Y. IoT-enabled adaptive systems for dynamic worker allocation. *Int. J. Prod. Res.* **2022**, *60*, 3456–3470.
25. Gue, K.R.; Kim, H.H. An approximation model for sojourn time distributions in acyclic multi-server queueing networks. *Comput. Oper. Res.* **2012**, *63*, 46–55. [CrossRef]
26. Gue, K.R.; Kim, H.H. Predicting departure times in multi-stage queueing systems. *Comput. Oper. Res.* **2012**, *39*, 1734–1744. [CrossRef]
27. Kelton, W.D.; Sadowski, R.P.; Sadowski, D.A. *Simulation with Arena*; McGraw-Hill: New York, NY, USA, 2007.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.