

REPRESENTATIVE-DAY DISCRETE-EVENT SIMULATION TO REDUCE OVERTIME IN OUTBOUND WAREHOUSE DOCKING

Pachira Na nan¹, Thammawit Prasert² and Peerapong Pakawanich^{3*}

¹M.Eng. Student, Department of Industrial Engineering and Management, Faculty of Engineering and Industrial Technology, Silpakorn University, Mueang District, Nakhon Pathom, 73000, Thailand. Email: nanan_p2@su.ac.th

^{2,3*}Lecturer, Department of Industrial Engineering and Management, Faculty of Engineering and Industrial Technology, Silpakorn University, Mueang District, Nakhon Pathom, 73000, Thailand,

²prasert_t@su.ac.th, ³pakawanich_p@su.ac.th

**Corresponding author*

ABSTRACT

Overtime in outbound warehouse docking operations is a persistent logistics challenge, yet existing studies rarely address realistic labor calendars, zoning constraints, heterogeneous truck-type processes, or extreme congestion days. This study develops a data-driven, minute-resolution discrete-event simulation (DES) in Python to evaluate five dispatching and bay-sharing policies (S0–S4) at a Thai beverage manufacturer. Operational data from 235 working days covering four truck types (EXP, MAN, CAN, BOT) across 19 bays were used to calibrate the model. A hybrid scenario-reduction approach-combining tail-based extreme-day retention and Ward's hierarchical clustering-compressed the dataset to 40 representative days (including 16 extreme days), reducing computational effort by 83% while preserving workload fidelity (NRMSE = 0.070; Pearson r = 0.873). A total of 200 DES runs (40 days $\times$ 5 policies) were conducted, with all improvements confirmed statistically significant (paired t-test, p < 0.001). FIFO dispatching with BOT-to-CAN bay sharing (S1) proved most robust, reducing weighted-average overtime by 43.34% (from 112.94 to 63.99 min/day) with the lowest truck waiting times. While S2 achieved slightly lower absolute overtime (62.79 min/day), it redistributed delays under peak conditions rather than relieving congestion. Sensitivity analysis identified a 90-minute end-of-shift activation window as optimal. S1 is recommended as the operational default, deployable via gate-level queue management

without optimization infrastructure. The proposed DES framework offers an efficient, tail-aware platform for evidence-based dock policy evaluation before real-world implementation.

KEYWORDS: discrete-event simulation, outbound logistics, overtime reduction, representative-day selection, truck dock scheduling.

1. Introduction

Transportation and logistics operations play a central role in ensuring supply chain continuity, particularly at the plant interface where inbound and outbound flows are consolidated. In such environments, dock and bay management significantly affects operational performance from truck arrival through docking, loading, and departure. Ineffective dock control can propagate congestion, prolong waiting times [1-5], and ultimately lead to overtime, making dock operations a persistent managerial concern in logistics and manufacturing systems [6].

A substantial body of research models dock-level decision-making as door or dock assignment and truck scheduling problems [2-4,7,8]. Prior studies show that assignment and sequencing decisions made under limited dock capacity and handling resources strongly influence throughput, waiting time, and service performance [1-3,7]. Optimization-based approaches dominate this literature, ranging from exact formulations to heuristic and metaheuristic methods, with several studies proposing integrated door–schedule models that incorporate handling times and operational interactions [1,6,8]. For example, Li et al [9] developed an integrated model that jointly determines dock mode configuration, truck assignment, and scheduling under mixed service mode docks where loading and unloading operations share dock capacity. Their study highlights that flexible dock configurations can substantially improve operational performance compared with static assignment strategies. However, analytical optimization models often simplify operational details such as labor calendars, zoning rules, and dispatching practices. Consequently, they may have limited ability to explain how congestion accumulates throughout the day and eventually leads to overtime under realistic end-of-shift conditions.

To capture these operational complexities, discrete-event simulation (DES) has been widely applied in logistics and warehousing research. DES explicitly models event-driven interactions among trucks, docks, equipment, and labor resources, allowing operational rules and constraints to be represented directly. Reviews identify DES as a key methodology for analyzing logistics

systems where operational feasibility and timing interactions are difficult to represent analytically [10]. Previous studies demonstrate that DES can effectively evaluate trade-offs among dock utilization, waiting time, and resource allocation under alternative operating policies [11,12]. Simulation has also been incorporated into digital-twin frameworks, where operational data are used to calibrate models for monitoring and decision support [11]. Related work by Kaarlela et al [13] illustrates how simulation environments combined with data-driven models can support operational planning and scenario evaluation in transportation systems. Such perspectives are particularly relevant for overtime management because overtime depends not only on total workload but also on when congestion occurs relative to labor schedules and breaks [14].

Overtime should therefore be viewed not only as a cost indicator but also as an operational outcome influenced by staffing levels, break feasibility, and job-specific processing requirements. In many dock operations, different truck types require different processing structures, meaning that similar arrival volumes may generate very different labor workloads and congestion patterns. As a result, evaluating dispatching, sharing, and zoning policies requires models that explicitly represent labor calendars and shop-floor rules rather than relying solely on idealized capacity assumptions. Simulation-based policy evaluation is therefore well suited for comparing operational policies under realistic operating conditions.

A key challenge in simulation-based dock studies, however, concerns experimental design when long-term operational data are available. Many studies rely on "typical-day" scenarios or synthetic inputs that smooth out demand variability, even though overtime is often triggered by rare peak-congestion days rather than average conditions. To address this issue, representative-day selection—an approach related to scenario reduction in long time-series analysis—has been proposed as a practical way to preserve demand variability while reducing computational effort [15,16]. By explicitly retaining extreme days instead of averaging them, such methods can maintain low-probability but high-impact operating conditions that strongly influence overtime formation [6,15,16].

In representative-day frameworks, each operating day is encoded using features that capture workload level and variability, such as daily truck volume and temporal indicators. A hybrid "extreme + typical" strategy is commonly applied, where extreme days are first identified and preserved to retain rare operational conditions [17-19]. The remaining days are then clustered to identify representative normal scenarios using clustering techniques such as hierarchical clustering, with the number of clusters determined through internal

validation metrics [15,20,21]. Representative days selected from each cluster can subsequently be used to reconstruct annual workload patterns and evaluate operational policies while maintaining fidelity to the original time-series characteristics [15,16,22].

In practice, recurring overtime has been observed in outbound dock operations where dispatching decisions are performed manually and non-FIFO sequencing is frequently applied. Trucks arriving later may be served earlier, resulting in long and uneven waiting times and raising concerns regarding fairness and operational governance. The company therefore seeks to evaluate whether FIFO-based dispatching, potentially combined with controlled bay sharing, can reduce driver waiting without increasing overtime under realistic operational constraints.

Despite extensive research on dock assignment, truck scheduling, and simulation-based analysis, important gaps remain for this decision context. Existing studies often evaluate system performance under stylized or average operating conditions, while extreme congestion days that strongly influence overtime are underrepresented. In addition, many optimization-based approaches are tested under simplified problem settings that do not explicitly represent labor calendars, zoning restrictions, and heterogeneous job processes. As a result, limited attention has been given to governance-oriented dispatching policies such as FIFO implemented under realistic operational constraints, and the interaction between fairness-driven rules and overtime formation remains insufficiently understood.

Motivated by these gaps, this study develops a data-calibrated, minute-resolution discrete-event simulation (DES) based on real operational data from an outbound warehouse. A representative-day selection procedure is used to preserve annual workload variability, including extreme operating conditions, while keeping the experimental design computationally tractable. Within this framework, alternative dispatching and controlled sharing policies are evaluated under identical arrival streams to quantify their impacts on waiting time and overtime. The results provide evidence-based guidance for selecting governance-aligned dock policies prior to real-world implementation.

2. Methodology

2.1 Case Study System

The case study focuses on a large beverage manufacturer's outbound transportation unit in Thailand, where finished goods are loaded into third-party carrier trucks via 19 loading

bays, which include four job types: export (EXP), manual/hand-stacked (MAN), canned products (CAN), and bottled products (BOT), divided into three operational zones: bays 1-6 serve EXP and MAN, bays 7-8 serve CAN, and bays 9-19 serve BOT, Table 1 outlines the process structure and projected resource requirements for each job type at the outgoing port. Each row includes the truck type (EXP, MAN, CAN, BOT), the number of consecutive steps required for completion, the step labels that describe the DES workflow (loading, positioning), the time (minutes), and the number of workers (people) assigned to each step. EXP and MAN are designed as two-step processes. Step 1 involves a forklift loading goods onto a truck with two employees, followed by Step 2 where another group of three employees positions the goods. Resources are shared between EXP and MAN, using the same dock and labor. However, Step 2 of EXP is significantly longer than MAN ($20 + 66 = 86$ minutes vs. $20 + 26 = 46$ minutes). CAN is the simplest and fastest unloading method, involving only one step: the forklift loading goods onto the truck with two employees, with a loading time of 12 minutes per job. BOT is also a two-step process. Step 1 involves the forklift loading goods onto the truck with two employees (25 minutes for two employees per unloading), followed by the necessary rest period (20 minutes). The truck owner is responsible for covering the goods for transport. This area is already occupied, so the forklift operator can work on unloading goods at another dock for the BOT job.

Table 1 Work-type process structure and nominal service times

Work type	Stage description	Distribution	Parameters	Number of workers
EXP	Loading	Normal	$N(20, 7.12^2)$	2
	positioning	Triangular	$T(46, 66, 86)$	3
MAN	Loading	Normal	$N(20, 7.12^2)$	2
	positioning	Normal	$N(26, 6^2)$	3
CAN	Loading	Empirical	Empirical data with mean = 12	4
BOT	Loading	Empirical	Empirical data with mean = 25	10
	Mandatory hold	Deterministic	Fixed = 20	-

2.2 Motivation for Observed Operations

According to dock logs, around 65% of demand is accounted for by BOT, followed by CAN (28%), and the remainder by EXP and MAN. Although the port is designated, bay sharing is used in practice to alleviate congestion. CAN bays are occasionally utilized by BOT (around 2-3% overall, peaking to ~14% on congested days), and a considerable share of CAN vehicles is processed in bays 1-6 (about 15% total). It is suggested by these patterns that congestion is relieved through discretionary, experience-driven reassignment rather than by a constant rule whereby some bottlenecks are eliminated while other flows are interfered with. This effort is driven by the operational concern of late-day queue growth and overtime, which is frequently caused by concentrated arrivals (particularly around midday) combined with zoning, team, and calendar limitations. The DES is thus designed as a realistic testbed for present practice (S0) to be compared with various dispatching and controlled sharing policies (S1-S4) under identical arrival streams.

2.3 Data collection and preparation.

Operational data were collected from a beer distribution warehouse over 235 working days in 2024, covering four truck types: Export (EXP), Manual (MAN), Canned (CAN), and Bottled (BOT). The dataset included truck arrival timestamps, dock check-in times, and loading start and finish times. Inter-arrival times were derived from consecutive arrival timestamps of each truck type. Since arrivals are unscheduled and driver-dependent, empirical distributions constructed from the 235-day records were used for all four truck types. Service times were fitted using the Kolmogorov–Smirnov test ($\alpha = 0.05$); parametric distributions were accepted when $p > 0.05$, otherwise empirical distributions were applied. All resulting distributions and parameters are summarized in Table 1. Operational rules that are not fully recorded in digital systems are elicited through stakeholder interviews and encoded as model constraints, including BOT adjacency, team rules, and bay usage restrictions. Days with no activity, duplicate records, and inconsistent timestamps are removed during the data cleaning process.

2.4 Representative-Day selection.

Each day is encoded as a feature vector capturing workload level and variability (daily truck volume, 7-day rolling mean, 7-day rolling standard deviation) plus calendar indicators (day-of-week, month, and weekend/holiday flags when available). A hybrid “extreme + typical” representative-day approach is then applied to reduce the dataset without losing rare congestion/underload behavior. Extreme days are identified via tail screening of workload indicators and anomaly signals from SOM quantization error, and then retained (16 days) to preserve low-probability, high-impact conditions. The remaining days are clustered using Ward’s hierarchical clustering, with the number of clusters selected using the silhouette coefficient. Within each cluster, representative normal days are chosen using stratified quantile sampling and centroid/medoid-style selection to maintain within-cluster fidelity. Representativeness is evaluated by reconstructing the annual workload series using closest-match assignment and representative-day weights, supported by NRMSE and Pearson correlation as standard reconstruction accuracy metrics.

Figure 1 Duration-curve overlay (actual vs. reconstructed annual workload). The empirical duration curve (blue) is closely tracked by the reconstructed curve from the representative-day set (orange) across both typical days and extremes, indicating high distributional fidelity (NRMSE = 0.070). Figure 2 Time-series overlay (actual vs. reconstructed annual workload). Daily shipment volumes over the 2024 calendar year (x-axis) are compared between observed data (blue) and the representative-day reconstruction (orange). The overall temporal pattern and major fluctuations are reproduced by the reconstruction, showing strong agreement in trend/shape (Pearson $r = 0.873$).

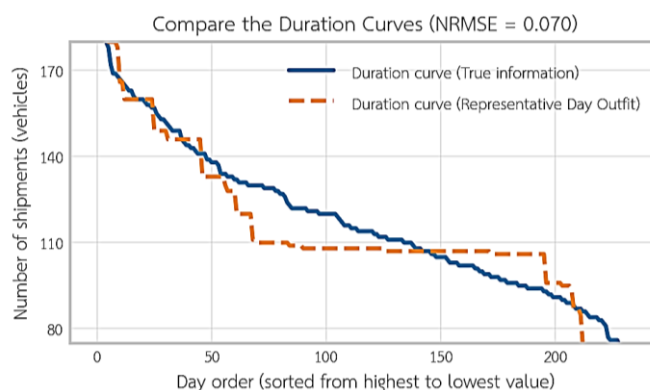


Figure 1 Duration-curve overlay actual vs reconstructed

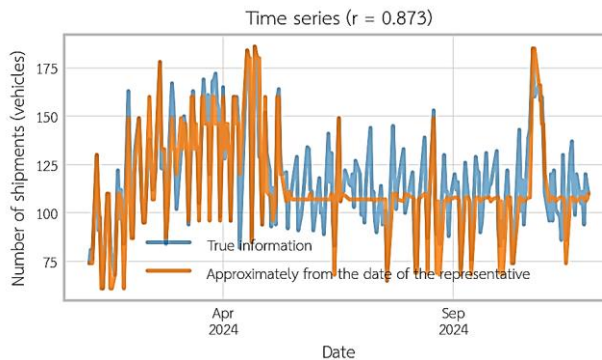


Figure 2 Time-series overlay actual vs reconstructed

Regarding benchmark workload regimes for scenario design, to standardize comparisons in the experimental phase, three benchmark days are defined as anchors for scenario construction: MAX (high-load stress test), AVG (typical-load), and MIN (low-load reference).

Table 2 Benchmark days for scenario design.

Regime	Benchmark day (date)	Daily truck count	Selection rationale
MAX	2024-04-22	196	Stress-test day
AVG	2024-02-07	107	Typical regime
MIN	2024-09-06	68	Low-load reference

Figure 3 summarizes the representative-day selection workflow used to reduce 235 working days to a 40-day experimental set while preserving both typical variability and rare extremes. Each day is encoded as a feature vector, and extreme days are detected using tail-based workload screening and SOM quantization error, yielding 16 extreme days. The remaining 219 non-extreme days are clustered with Ward's hierarchical clustering; $k = 3$ clusters are selected and treated as distinct workload regimes: C0 ($n = 157$, mean = 108.2 trucks/day, std = 17.0) represents the dominant normal operating regime with moderate daily volumes and the widest workload variability; C1 ($n = 55$, mean = 146.5 trucks/day, std = 15.5) captures a high-load regime in which even the lowest-volume day exceeds the median of C0; and C2 ($n = 7$, mean = 105.6 trucks/day, std = 10.0) represents a rare low-variance regime characterized by near-average volumes with unusually stable daily throughput. Twenty-four normal representative days

(C0 = 17, C1 = 6, C2 = 1) are then selected via stratified quantile and centroid-based sampling in proportion to cluster size, and are merged with the 16 extreme days to form the final 40-day experimental set. Representativeness is validated by weighted annual reconstruction (NRMSE = 0.070; $r = 0.873$; Figures 1-2); if acceptable, the representative set is used for simulation.

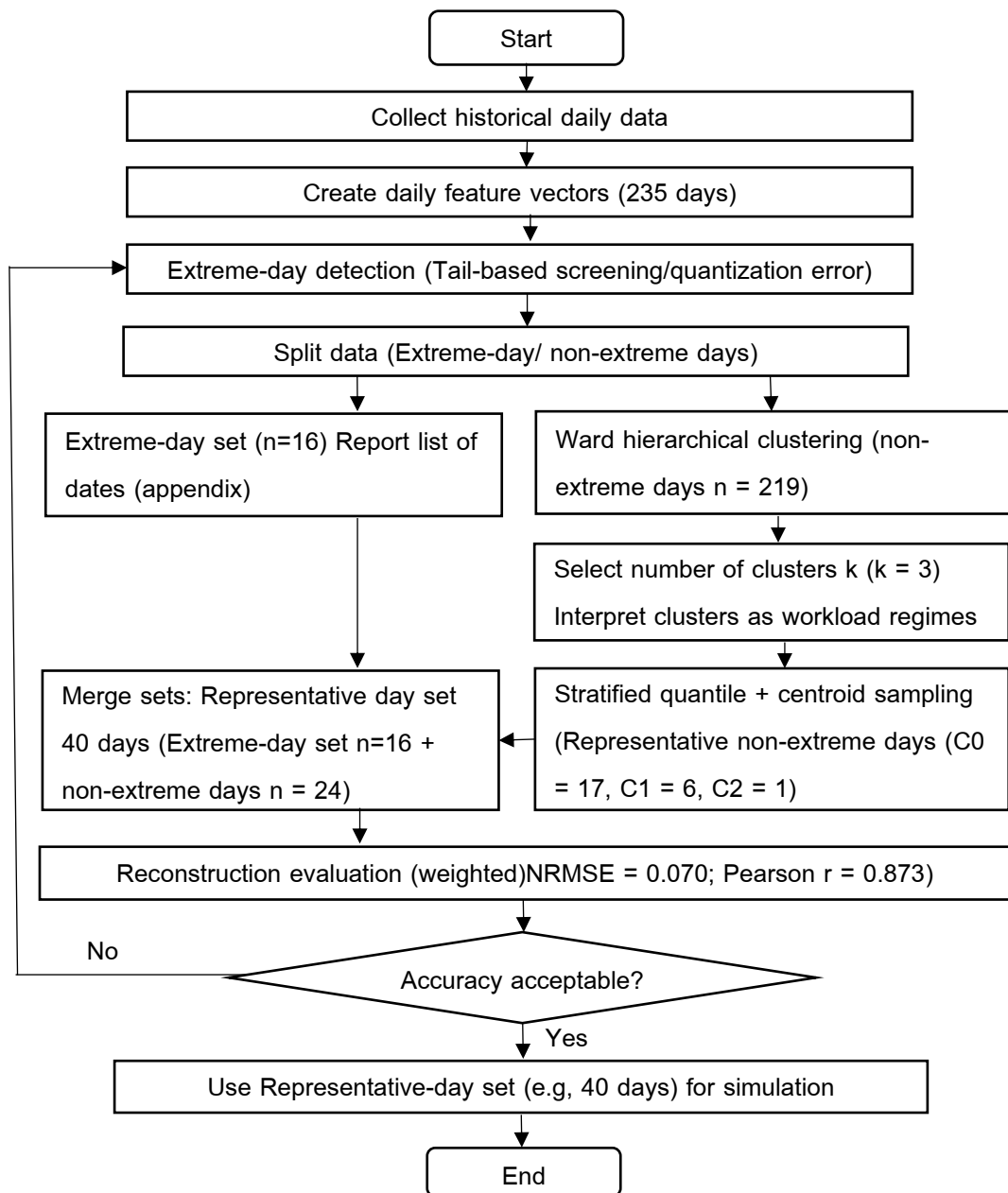


Figure 3 Representative-day selection procedure and outputs.

2.5 DES model and truck dispatching support system control logic.

The discrete-event simulation was implemented in Python (Visual Studio Code; 64-bit, 4-core 2.8 GHz CPU, 16 GB RAM). Model inputs were parameterized using distributions reported in Table 1. Truck arrivals were generated from empirical inter-arrival-time distributions derived from 235 operating days, and each truck progressed through sequential alignment and loading stages, with service times sampled from the corresponding distributions. BOT trucks incurred an additional 20-minute post-loading detention per operating rules, and resources were allocated on a first-come, first-served basis at the staffing levels specified in Table 1.

To capture annual demand variability, 40 representative days spanning the full range of operating conditions were identified from historical records. Each scenario (e.g., S1, S2) was evaluated by running the simulation across all 40 representative days, with each day's empirical arrival profile applied as a distinct model input rather than as a stochastic replication. Because service-time distributions for high-volume truck types were tightly concentrated around their means (CAN: $n = 2,239$; BOT: $n = 6,965$), output variability attributable to service-time sampling was negligible, and no additional replications per day were required. Each run simulated one complete operating day, initialized from an empty system at 08:00 and terminated upon departure of the last truck; no warm-up period was required.

2.6 Policies and experimental design.

This section specifies the operating policies evaluated in the event-driven DES (Figure 4). The simulation is implemented in Python and parameterized from historical operations, encompassing arrival patterns, service-time distributions, shift constraints, and time-varying resource availability. Truck state attributes—including urgent/non-urgent status, BOT/CAN classification, and (when enabled) slow/fast labels derived via recursive partitioning (RPart)—are carried throughout the simulation. The DES logic is held constant across all experiments; policy effects are isolated exclusively to the dispatch decision step, where candidate ranking and BOT–CAN sharing permissions are applied subject to zoning constraints and resource calendars.

Input:

Day Data: trucks i with (id, arrival time, type)

Bays by zone; process structures and times; resource calendars (shift/lunch);
operational constraints (zoning, BOT team/adjacency, etc.)

Output:

Per-truck metrics (start, finish, wait, flow) and daily KPIs (OT, mean waits)

Procedure RUN_DES (Day Data):

Sort trucks by (arrival time, id); $idx \leftarrow 1$

$t \leftarrow 0$

Initialize bay state (FREE), resource state from calendars

Initialize FIFO queues Q_EXP , Q_MAN , Q_CAN , $Q_BOT \leftarrow$ empty

Initialize event list $E \leftarrow$ empty (min-heap by time)

ENQUEUE_ARRIVALS(t); DISPATCH(t) // initial starts

while ($idx \leq |Day\ Data|$) or (E not empty) do

$t \leftarrow \min(\text{next_arrival_time}(idx), \text{next_event_time}(E))$

Update resource_state at time t (shift/lunch feasibility)

ENQUEUE_ARRIVALS(t)

while (E has events at time t) do

$e \leftarrow \text{pop}(E)$

PROCESS_EVENT(e, t) // release resources, stage transitions, free bays

end while

DISPATCH(t)

end while

COMPUTE_KPIS ()

return outputs

Sub procedure ENQUEUE_ARRIVALS(t):

while ($idx \leq |Day\ Data|$) and ($Day\ Data[idx].\text{arrival time} \leq t$) do

$i \leftarrow Day\ Data[idx]$

push i into FIFO queue $Q_type(i)$

$idx \leftarrow idx + 1$

end while

Figure 4 Event-Driven DES with TDSS

```

Sub procedure DISPATCH(t):
  // Step 1: choose candidates from each queue using FIFO order (baseline order)
  // Step 2: start feasible jobs subject to bays, resources, and constraints
  START_EXP_MAN(t)
  START_CAN(t)
  START_BOT(t)// includes BOT team + adjacency constraints
  // Note: sharing/priority rules are NOT applied here (handled by policy blocks)
Sub procedure PROCESS_EVENT (e, t):
  Release resources tied to completed stage
  if (truck has next stage) then
    Start next stage when feasible, else mark as waiting-for-stage
    Schedule its completion event in E
  else
    Mark truck completed; free bay
  end if
Sub procedure COMPUTE_KPIS ():
  For each truck i:
    wait_i ← start_i — arrival_i
    flow_i ← finish_i — arrival_i
  day_finish ← max_i(finish_i)
  OT ← max (0, day_finish — SHIFT_END)
  Report mean waits by type and OT

```

Figure 4 Event-Driven DES with TDSS (continued)

Five policies (S0–S4) are evaluated by varying only the dispatch rule set. S0 represents current practice (non-FIFO, type separation, urgent-first prioritization, with urgent BOT trucks permitted CAN access when feasible). S1 enforces FIFO with BOT-to-CAN sharing. S2 extends S1 by introducing a tail-of-shift rule that prioritizes slow tasks during the final 90 minutes before shift end. S3 applies the same tail-of-shift framework but prioritizes fast tasks instead. S4 evaluates protected capacity pooling, in which BOT-to-CAN sharing is permitted only when CAN operations are not adversely affected.

The experiment follows a repeated-measures design in which all five policies are applied to each of the 40 representative days, yielding 200 DES runs (40 days $\times$ 5 policies). Since the same empirical arrival stream is used across policies within each day, day-to-day demand variation is controlled and policy effects are attributed solely to the dispatch rule. This paired-by-day structure forms the basis for subsequent statistical comparisons.

2.6.1 Scenario 0: Baseline TDSS S0—current practice

The as-is benchmark policy is defined by Scenario 0. Observed practice is followed for dispatch specifically, non-FIFO, type-separated sequencing with urgent-first overrides where CAN bays may be used by urgent tasks when feasible and capacity is available. In the DES, all core mechanics are fixed (arrivals via `ENQUEUE_ARRIVALS(t)` and resource release/stage transitions via `PROCESS_EVENT(e,t)`); policy influence is restricted to `DISPATCH(t)`, where feasibility (zoning, resource calendars, and BOT team/adjacency constraints) is enforced and the baseline priority logic is then applied. As summarized in Figure 4, no additional end-of-shift prioritization beyond feasibility checks is included in S0.

2.6.2 Alternative Dispatching Policies (S1–S4)

In Scenario 1 (S1), FIFO dispatching with BOT-to-CAN sharing is introduced to examine whether BOT congestion and overtime can be reduced relative to S0. Trucks are served in FIFO order within each type queue under the same feasibility constraints. When BOT demand exists but no BOT bay is available, a BOT truck may use an idle CAN bay (7–8) if sharing conditions are satisfied.

Scenarios 2 and 3 (S2–S3) extend S1 by introducing a tail-of-shift priority rule to examine whether overtime is influenced by end-of-shift sequencing. Outside the final window, dispatch follows the S1 logic. Within the last $W = 90$ minutes before `SHIFT_END`, feasible candidates are re-ranked using slow/fast task labels, where S2 prioritizes slow tasks and S3 prioritizes fast tasks, with FIFO used as the tie-break rule.

In Scenario 4 (S4), protected capacity pooling is evaluated. FIFO dispatching is retained, but BOT-to-CAN sharing is allowed only when CAN operations are unaffected. Specifically, CAN bays may serve BOT trucks only when protection conditions are satisfied

(e.g., no CAN queue or the CAN bay has remained idle for at least τ minutes); otherwise, BOT trucks remain restricted to BOT-eligible bays.

2.7 Performance measures and validation.

Daily overtime in minutes and mean waiting times for BOT and CAN trucks are defined as the primary performance indicators. To support the interpretation of policy effects, additional mechanism indicators are recorded, including the frequency of BOT to CAN sharing and the proportion of BOT trucks processed in CAN bays. Model verification is conducted via deterministic sanity tests covering zoning rules, first-in-first-out behavior and policy overrides, lunch and shift feasibility, and BOT overflow restrictions. For model validation, simulated outputs are compared with historical patterns for selected benchmark days, and stakeholder review is incorporated to confirm that operational practice is reflected by the encoded rules and modeled behaviors.

Prior to hypothesis testing, normality of the paired differences ($S_i - S_0$) for each KPI (OT, BOT mean wait, CAN mean wait) was assessed using the Shapiro–Wilk test ($\alpha = 0.05$), which is recommended for small samples ($n = 40$). Where normality was not rejected ($p > 0.05$), a paired t-test was applied; otherwise, the Wilcoxon signed-rank test was used as a non-parametric alternative.

3. RESULTS AND DISCUSSION

3.1 Representative-day design enables efficient, variability-preserving experiments

The selection pipeline reduced the experimental set from 235 working days to 40 representative days (including extremes) while preserving annual workload structure. Reconstruction accuracy remained high (NRMSE = 0.070; Pearson $r = 0.873$), indicating that typical patterns and tail congestion are retained. This reduces evaluation effort by $\approx 83\%$ and supports a fully crossed design of 200 DES runs (40 days $\times$ 5 policies) with paired-by-day comparisons. Table 3 reports the regime composition in the original year and in the representative set, together with the annual weights used to compute the weighted-overall Key Performance Indicator (KPIs).

Table 3 Day-type composition and annual weights

Day type	Actual days	Representative days	Weight
MAX	10	6	0.04
MIN	26	10	0.11
AVG	199	24	0.85

3.2 Model credibility on benchmark days verification and validation.

The model's reliability was evaluated by comparing observed and simulated overtime (OT) on three benchmark days used to anchor the MAX/AVG/MIN workload regimes. As shown in Table 4, the expected congestion ordering ($MAX > AVG > MIN$) is reproduced by the DES, and historical OT on all three benchmarks is roughly matched. Based on the usual relative error definition from the findings, a tiny deviation (+0.42%, i.e., 241 vs 240 minutes) is exhibited by the MAX benchmark, although the AVG and MIN are found to be identical (0.00%). It is suggested by this close agreement that the model's stored calendars and operational restrictions (such as shift/lunch feasibility and dock zoning logic) are congruent with observed completion behavior for benchmark regime anchors. In following experiments, paired-by-day policy comparisons (same arrival stream and limitations, different rules) continue to be emphasized, because reliable inference on rule impacts is provided even when unobserved micro-disruptions or discretionary floor judgments vary outside of the recorded data.

Table 4 Benchmark-day overtime validation.

Benchmark date	Regime	OT observed (min)	OT simulated (min)	% error*
2024-04-22	MAX	240.0	241.0	+0.42%
2024-02-07	AVG	78.6	78.6	0.00%
2024-09-06	MIN	22.0	22.0	0.00%

3.3 Simulation results

Distinct performance profiles for each operational strategy are revealed by the comprehensive evaluation of Table 5. The current practice (S0) is established as the baseline

of inefficiency, where the highest overtime and excessive delays for all truck types are exhibited. The most robust system improvement is delivered by the transition to S1, with weighted overtime being reduced by 43.34% and the lowest waiting times secured; notably, S1's superior robustness across different load regimes is confirmed by Figure 5, as significant service degradation is avoided even under maximum load regimes. While the absolute minimum overtime is achieved by S2, substantial risk is introduced, as evidenced by a sharp spike in waiting times for CAN truck types during peak congestion (Figure 5c). Among the remaining alternatives, S3 presents a relatively weaker overall weighted profile despite specific relief for BOT delays being offered during peaks, whereas overtime is successfully controlled by S4 but a severe service imbalance is created by disproportionately increased delays for BOT truck types. Consequently, S1 is identified as the optimal primary strategy, with the others being regarded as specialized alternatives with narrower operational utility with more limited and context-specific operational applicability.

Table 5 Weighted-overall KPIs annualized mean per day.

KPIs	S0	S1	S2	S3	S4
OT (min/day)	112.94	63.99	62.79	65.07	63.84
BOT mean wait (min)	134.68	27.55	34.30	41.29	53.04
CAN mean wait (min)	85.20	20.41	23.35	26.52	21.97

Before conducting paired t-tests, the normality of the paired differences ($S_i - S_0$) for each KPI was assessed using the Shapiro–Wilk test ($\alpha = 0.05$). Since normality was not rejected, paired t-tests were applied to compare each policy with the baseline S0 using the same 40 representative days. For OT, BOT mean waiting time, and CAN mean waiting time, all comparisons showed statistically significant improvements over the baseline policy ($p < 0.001$). These results indicate that the improvements reported in Table 5 reflect systematic policy effects rather than random day-to-day variation.

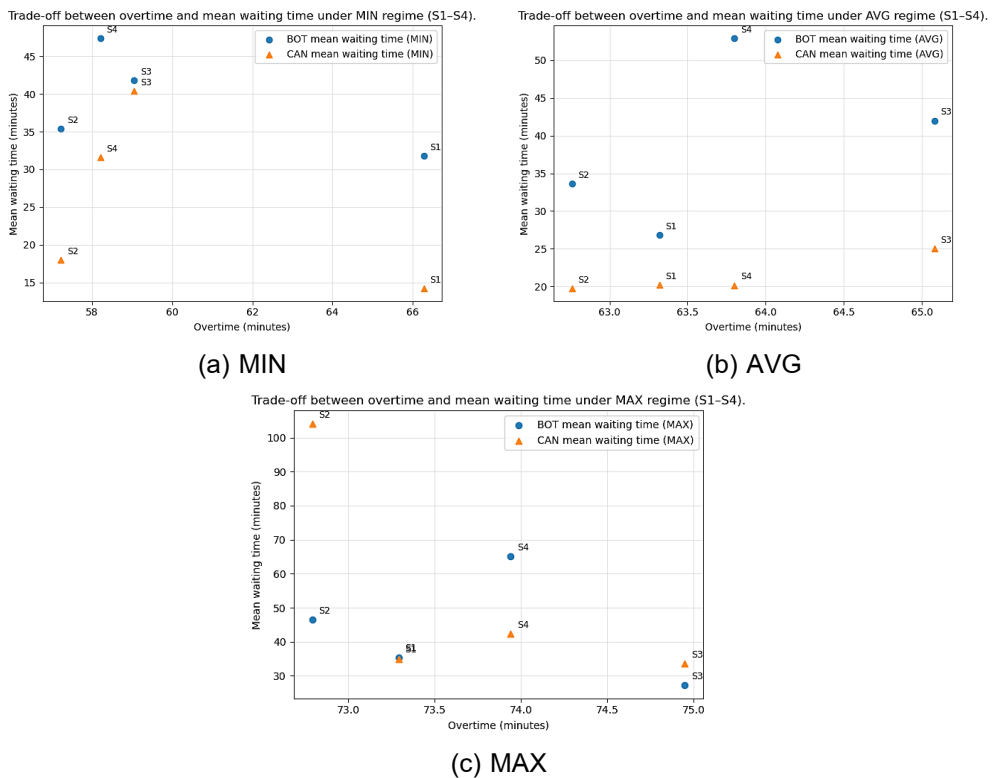


Figure 5 Trade-off between overtime and mean waiting time

3.4 Mechanism evidence from paired-by-day contrasts interpretable contributions.

To isolate true policy effects from day-to-day workload variation, paired-by-day contrasts are computed within each representative day and summarized by regime (MIN/AVG/MAX), as reported in Table 6. Across regimes, overtime is reduced by all scenarios relative to S0, confirming that system congestion rather than isolated sub-processes is addressed by the interventions; the lowest overtime is yielded by S2, but its advantage over S1 is small, suggesting diminishing returns once queues are stabilized by sharing. Substantial improvements in BOT waiting are observed under all policies, with the largest MAX-day reduction found under S3 (226.5 to 27.2 min) and S1 close behind (to 35.4 min), indicating that BOT queue buildup is effectively suppressed by both tail-of-shift logic and FIFO + sharing during peak congestion. For CAN waiting, consistently strong reductions across regimes are delivered by S1, and the best performance under MAX (137.4 to 33.6 min) is exhibited by S3, whereas a clear peak-day drawback is shown by S2 with CAN waiting

remaining high under MAX (104.1 min), implying that some overtime gains are achieved by shifting delay toward CAN in extreme conditions. Overall, queue stabilization via FIFO with controlled form BOT to CAN sharing (S1) is identified by the contrasts as the most robust default, while S2 is best treated as a selective option when potential losses in CAN reliability under maximum-load days are outweighed by overtime minimization.

Table 6 Paired contrasts by day type.

Day type	S0 (min)	S1 (min)	S2 (min)	S3 (min)	S4 (min)
(a) Overtime (OT)					
MIN	122.4	66.28	57.22	59.05	58.2
AVG	111.5	63.32	62.76	65.08	63.8
MAX	123.86	73.29	72.79	74.95	73.94
(b) BOT mean waiting time					
MIN	90	31.8	35.4	41.8	47.4
AVG	133.8	26.8	33.6	41.96	52.91
MAX	226.5	35.4	46.4	27.2	65.1
(c) CAN mean waiting time					
MIN	103.8	14.2	18	40.4	31.6
AVG	81	20.2	19.73	25	20.14
MAX	137.4	34.99	104.1	33.6	42.3

3.5 Managerial implications and policy deployment guidelines

The finding that FIFO with controlled cross-zone sharing (S1) consistently outperforms other configurations across MIN, AVG, and MAX regimes aligns with the simulation-based evidence reported by Azab et al [1], who demonstrated that structured appointment and sequencing policies reduce truck queuing and terminal congestion more effectively than unregulated or fully flexible alternatives. However, while Azab et al [1] focused on container terminal contexts where external truck arrivals are stochastic and appointment windows are predefined the present study extends this logic to an outbound warehouse docking environment where zone-level sharing rules and tail-of-shift decisions interact dynamically

with demand variability captured through representative-day clustering (MIN/AVG/MAX). This distinction is important: our results show that the relative advantage of S1 over S2 and S3 is not uniform across demand regimes—S2 gains merit under MAX conditions specifically for overtime suppression, a nuance not captured in single-scenario optimization frameworks such as that of Azab et al [1].

For practical implementation, managers should adopt S1 as the operational default by deploying a gate-level queue management system that records truck arrival timestamps and enforces service sequencing accordingly. This approach is consistent with Xu et al [5], who demonstrated that dynamic appointment rescheduling under arrival time uncertainty can reduce overall operating costs and alleviate port congestion—suggesting that structured, time-aware sequencing at the gate level supports both efficiency and managerial decision-making. Beyond sequencing governance, the tiered policy framework (S1 as default, S2/S3 as conditional overrides) can be operationalized through appointment-window parameterization in existing dock management systems—an approach consistent with Azab et al [1], who demonstrated that structured truck appointment policies reduce terminal congestion without requiring dedicated optimization infrastructure.

3.6 Sensitivity Analysis of Tail-of-Shift Window (W) for Scenarios S2 and S3

To examine the robustness of the tail-of-shift prioritization rules in S2 and S3, a sensitivity analysis was conducted by varying the activation window W at 60, 90, and 120 minutes before SHIFT_END. The baseline configuration used in the main experiments is $W = 90$ minutes. All other model components, including FIFO dispatching, BOT-to-CAN sharing rules, and resource calendars, remain unchanged. Each configuration was evaluated across the same 40 representative days, resulting in 240 additional DES runs (40 days $\times$ 3 window settings $\times$ 2 policies). Weighted-average KPIs are summarized in Table 7 for comparison with the main results.

The results indicate a non-monotonic relationship between window width W and overtime performance. For S2, reducing the window to $W = 60$ slightly increases overtime (64.12 minutes /day), suggesting insufficient time to reorder slow BOT trucks before shift end. Conversely, extending the window to $W = 120$ increases overtime to 67.43 minutes/day, exceeding even the S1 reference, due to early prioritization of slow trucks that reduces dock throughput and causes backlog accumulation. This effect is stronger in S3, where $W=120$

yields the highest overtime (70.81 minutes /day). Overall, the results indicate that $W = 90$ minutes provides the best performance balance, whereas larger windows may degrade system throughput. These findings confirm that $W = 90$ minutes represents a meaningful inflection threshold beyond which the prioritization rule transitions from beneficial to detrimental, and should be monitored carefully when calibrating the policy in practice.

Table 7 Sensitivity of S2–S3 performance to tail-of-shift window W

	W (min)	OT (min/day)	BOT mean wait (min)	CAN mean wait (min)
S1 (reference)	—	63.99	27.55	20.41
S2	60	64.12	32.18	21.84
S2	90	62.79	34.30	23.35
S2	120	67.43	35.12	31.07
S3	60	66.03	38.52	24.17
S3	90	65.07	41.29	26.52
S3	120	70.81	42.63	33.18

4. CONCLUSION

This study develops a Representative-Day Discrete-Event Simulation (DES) framework to reduce overtime in outbound warehouse docking operations at a Thai beverage manufacturer. By compressing 235 working days into 40 representative days using a hybrid extreme-retention and Ward's hierarchical clustering approach, the framework preserves annual workload variability while reducing computational effort by approximately 83% (NRMSE = 0.070; $r = 0.873$). Three benchmark regimes, MIN (68 trucks), AVG (107 trucks), and MAX (196 trucks), serve as workload-based decision contexts with annual weights of 0.11, 0.85, and 0.04, respectively.

Five dispatching and bay-sharing policies (S0–S4) were evaluated across all representative days in 200 DES runs. Paired t-tests ($\alpha = 0.05$) confirmed that all policies significantly outperform the current non-FIFO baseline (S0) ($p < 0.001$).

Results show that S1 (FIFO dispatching with controlled BOT-to-CAN sharing) is the most robust policy across regimes. Under AVG conditions, which represent most operating days, S1 reduces weighted-average overtime by 43.34% (112.94 to 63.99 min/day) while maintaining

the lowest waiting times in BOT (27.55 min) and CAN (20.41 min) zones. Under MIN conditions, S1 avoids unnecessary cross-zone interference, while under MAX conditions it remains the reliable baseline, with S3 suitable for peak-day BOT congestion and S2 applicable when overtime reduction is prioritized despite higher CAN waiting times. Sensitivity analysis indicates that a 90-minute tail-of-shift activation window provides the best performance balance.

The study extends representative-day selection methods to warehouse logistics, introduces regime-based policy evaluation using benchmark-day anchors, and distinguishes genuine overtime reduction from delay redistribution across zones. The framework also provides practical deployment guidelines without requiring full-year simulations, with validation errors close to zero across benchmark days.

Future research may explore real-time adaptive dispatching using machine learning based rule switching, integration with digital-twin systems for continuous monitoring, incorporation of cost-based objectives balancing overtime and service equity, and application of the representative-day DES framework to multi-facility or port terminal environments.

References

- [1] Azab A, Karam A, Eltawil A. A simulation-based optimization approach for external trucks appointment scheduling in container terminals. *International Journal of Modelling and Simulation* 2020;40(5):321-38. doi: 10.1080/02286203.2019.1615261.
- [2] Gelareh S, Glover F, Guemri O, Hanafi S, Nduwayo P, Todosijević R. A comparative study of formulations for a cross-dock door assignment problem. *Omega* 2020;91:102015. doi: 10.1016/j.omega.2018.12.004.
- [3] Monemi RN, Gelareh S. Dock assignment and truck scheduling problem; consideration of multiple scenarios with resource allocation constraints. *Computers & Operations Research* 2023;151:106074. doi: 10.1016/j.cor.2022.106074.
- [4] Shahmardan A, Sajadieh MS. Truck scheduling in a multi-door cross-docking center with partial unloading–reinforcement learning-based simulated annealing approaches. *Computers & Industrial Engineering* 2020;139:106134. doi: 10.1016/j.cie.2019.106134.
- [5] Xu B, Liu X, Li J, Yang Y, Wu J, Shen Y, et al. Dynamic appointment rescheduling of trucks under uncertainty of arrival time. *Journal of Marine Science and Engineering* 2022;10(5):695. doi: 10.3390/jmse10050695.

- [6] Teichgraeber H, Lindenmeyer CP, Baumgärtner N, Kotzur L, Stolten D, Robinius M, et al. Extreme events in time series aggregation: a case study for optimal residential energy supply systems. *Applied Energy* 2020;275:115223. doi: 10.1016/j.apenergy.2020.115223.
- [7] Tadumadze G, Emde S. Loading and scheduling outbound trucks at a dispatch warehouse. *IIE Transactions* 2022;54(8):770-84. doi: 10.1080/24725854.2021.1983923.
- [8] Sayed SI, Contreras I, Diaz JA, Luna DE. Integrated cross-dock door assignment and truck scheduling with handling times. *Top* 2020;28:705-27. doi: 10.1007/s11750-020-00556-z.
- [9] Li Y, Mohammadi M, Zhang X, Lan Y, van Jaarsveld W. Integrated trucks assignment and scheduling problem with mixed service mode docks: a Q-learning-based adaptive large neighborhood search algorithm. *European Journal of Operational Research* 2025;33(1):117-37. doi: 10.1016/j.ejor.2025.12.036.
- [10] Bottani E, Casella G. Discrete-event simulation in logistics and supply chain management: a scientometric perspective. *Production & Manufacturing Research* 2024;12(1):2415038. doi: 10.1080/21693277.2024.2415038.
- [11] da Silva MR, Agostino IR, Frazzon EM. Integration of machine learning and simulation for dynamic rescheduling in truck appointment systems. *Simulation Modelling Practice and Theory* 2023;125:102747. doi: 10.1016/j.simpat.2023.102747.
- [12] Derhami S, Smith JS, Gue KR. A simulation-based optimization approach to design optimal layouts for block stacking warehouses. *International Journal of Production Economics* 2020;223:107525. doi: 10.1016/j.ijpe.2019.107525.
- [13] Kaarlela T, Alekseev S, Maljamäki P, Ikäheimo E, Kurvinen E. Digital twin framework for electric truck route and infrastructure planning in the Nordic region. *Transportation Engineering* 2026;23:100426. doi: 10.1016/j.treng.2026.100426.
- [14] Rizqi ZU, Chou SY, Cahyo WN. A simulation-based digital twin for smart warehouse: towards standardization. *Decision Analytics Journal* 2024;12:100509. doi: 10.1016/j.dajour.2024.100509.
- [15] Cebecauer M, Jenelius E, Gundlegård D, Burghout W. Revealing representative day-types in transport networks using traffic data clustering. *Journal of Intelligent Transportation Systems* 2024;28(5):695-718. doi: 10.1080/15472450.2023.2205020.
- [16] Yeganefar A, Amin-Naseri MR, Sheikh-El-Eslami MK. Improvement of representative days selection in power system planning by incorporating the extreme days of the net

- load to take account of the variability and intermittency of renewable resources. *Applied Energy* 2020;272:115224. doi: 10.1016/j.apenergy.2020.115224.
- [17] Chen Z, Xiong Y, Yin B, Sun S, Zuo R. Recognizing geochemical patterns related to mineralization using a self-organizing map. *Applied Geochemistry* 2023;151:105621. doi: 10.1016/j.apgeochem.2023.105621
- [18] Li C, Conejo AJ, Sirola JD, Grossmann IE. On representative day selection for capacity expansion planning of power systems under extreme operating conditions. *International Journal of Electrical Power & Energy Systems* 2022;137:107697. doi: 10.1016/j.ijepes.2021.107697.
- [19] Licen S, Astel A, Tsakovski S. Self-organizing map algorithm for assessing spatial and temporal patterns of pollutants in environmental compartments: a review. *Science of The Total Environment* 2023;878:163084. doi: 10.1016/j.scitotenv.2023.163084.
- [20] Dalmaijer ES, Nord CL, Astle DE. Statistical power for cluster analysis. *BMC Bioinformatics* 2022;23(1):205. doi: 10.1186/s12859-022-04675-1.
- [21] Ribeiro R, Fanzeres B. Identifying representative days of solar irradiance and wind speed in Brazil using machine learning techniques. *Energy and AI* 2024;15:100320. doi: 10.1016/j.egyai.2023.100320.
- [22] Hoffmann M, Kotzur L, Stolten D, Robinus M. A review on time series aggregation methods for energy system models. *Energies* 2020;13(3):641. doi: 10.3390/en13030641.

Author's Profile



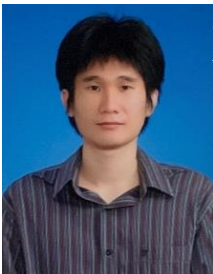
Pachira Na nan, M.Eng. Student at Department of Industrial Engineering and Management, Faculty of Engineering and Industrial Technology, Silpakorn University, Mueang District, Nakhon Pathom, 73000, Thailand. Email: nanan_p2@su.ac.th

Interested Research Area: Production planning and scheduling, Optimization, Logistics and Supply Chain Management



Thammawit Prasert, Lecturer at Department of Industrial Engineering and Management, Faculty of Engineering and Industrial Technology, Silpakorn University, Mueang District, Nakhon Pathom, 73000, Thailand. Email: prasert_t@su.ac.th

Interested Research Area: Design of Experiment, Production planning and scheduling, Optimization, Logistics and Supply Chain Management



Peerapong Pakawanich, Lecturer at Department of Industrial Engineering and Management, Faculty of Engineering and Industrial Technology, Silpakorn University, Mueang District, Nakhon Pathom, 73000, Thailand. Email: pakawanich_p@su.ac.th

Interested Research Area: Operations Research, Logistics and Supply Chain Management

Article History:

Received: January 21, 2026

Revised: April 18, 2026

Accepted: April 20, 2026