

Article

A Hierarchical Two-Level Adaptive Allocation Framework for Multi-Location Inbound Logistics Under Operational Constraints

Mohammad Hori * and Bernd Noche 

Department of Transport Systems and Logistics, Faculty of Engineering, University of Duisburg–Essen, Keetmanstraße 3–9, 47058 Duisburg, Germany; bernd.noeche@uni-due.de

* Correspondence: mohammad.hori-najafabadi@stud.uni-due.de

Abstract

Background: The allocation of non-divisible inbound deliveries across multiple warehouses requires the simultaneous consideration of capacity constraints, category-specific restrictions, workload balance, and long-term allocation consistency. **Methods:** This study proposes a hierarchical two-level allocation framework combining strategic category-rotation policies, normalized marginal scoring, and signed historical feedback. The algorithm is executed once per day to generate warehouse assignments for the following operational day. Historical correction is based on a rolling window covering the preceding 30 daily planning periods. **Results:** The framework was evaluated using daily simulation instances ranging from 100 to 1500 pallets, with an average of approximately 130 lots per pallet. Across all evaluated instances, the complete allocation procedure was completed in less than 5 s on the specified test system. The results indicate balanced warehouse utilization, progressive reductions in category–location imbalance, stable historical correction, and preservation of hard operational constraints. **Conclusions:** The framework provides an interpretable and computationally efficient approach for next-day inbound allocation. By combining explicit feasibility filtering, strategic policy signals, and a 30-day historical correction mechanism, it supports both short-term operational decisions and longer-term allocation balance.

Keywords: inbound logistics; warehouse allocation; hybrid optimization; adaptive allocation; decision support system; multi-location logistics; heuristic optimization; operational logistics

1. Introduction

Efficient allocation of inbound deliveries across multiple warehouse locations is a recurring operational challenge in contemporary supply chain management. Allocation decisions directly affect warehouse utilization, workload distribution, service continuity, and the ability of logistics networks to absorb fluctuating inbound volumes. The problem becomes particularly demanding when deliveries differ in pallet quantity, physical volume, product category, handling requirements, and divisibility, while warehouse locations simultaneously exhibit heterogeneous capacities and technical restrictions. In such environments, decisions must be generated within short operational time windows and must remain consistent across successive planning periods.

Research on supply chain allocation has traditionally focused on strategic and tactical problems. Foundational supply chain texts discuss network coordination, inventory positioning, and service-level management across interconnected facilities [1,2]. Classical inventory-planning approaches further address replenishment, production planning, and



Received: 9 June 2026

Revised: 14 July 2026

Accepted: 30 July 2026

Published: 6 August 2026

Copyright: © 2026 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 \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

inventory control in multi-location systems [3,4]. Facility location and network design models determine long-term capacity configurations and aggregate material flows [5,6], while advanced planning and supply chain management frameworks integrate procurement, production, distribution, and transportation decisions across several planning levels [7,8]. These approaches provide an important theoretical foundation, but they generally operate with aggregated or divisible flows and are therefore not directly designed for the daily assignment of discrete, non-divisible inbound deliveries to existing warehouses.

Mathematical programming has also been extensively applied to operational logistics assignment problems. Integer and linear optimization provide the formal basis for discrete assignment and capacity decisions [9,10], while stochastic programming, numerical optimization, and network-flow theory offer complementary tools for uncertainty, continuous parameterization, and flow-based constraints [11–13]. Mixed-integer linear programming formulations enable the explicit representation of capacity limits, assignment restrictions, time-related constraints, and multiple operational objectives. For example, real-time truck-to-door assignment at cross-docking facilities has been formulated as a mixed-integer scheduling problem to reduce inbound service and waiting times [14]. Although exact formulations provide globally optimal solutions for solvable instances, their computational burden may increase rapidly with the number of deliveries, locations, resource dimensions, and operational constraints. Moreover, repeated model solution may be difficult to integrate into decision environments in which warehouse states change after each assignment and recommendations must be explained to operational planners.

Heuristic and metaheuristic methods address part of this computational challenge. Surveys of multi-attribute routing and metaheuristic search emphasize the value of constructive and neighborhood-based procedures for complex logistics problems [15,16]. Adaptive large neighborhood search, for instance, modifies the selection frequency of search operators according to their historical search performance [17], while hybrid genetic and neighborhood-search procedures combine complementary search mechanisms to improve solution quality [18]. These methods adapt the algorithmic search process; however, their historical information is primarily used to select search operators, control diversification, or improve convergence. It does not normally represent persistent operational imbalance between shipment categories and warehouse locations.

Rolling-horizon approaches provide another mechanism for dynamic decision-making. They repeatedly update input information and re-solve a finite planning horizon as new tasks or system states become available. Recent warehouse research has combined rolling-horizon control with adaptive neighborhood search for dynamic task allocation in multi-AGV systems [19]. Such methods are well suited to dynamically arriving tasks, but their adaptation is primarily achieved by shifting and reoptimizing the planning horizon. The historical state is therefore incorporated through updated problem instances rather than through a persistent corrective signal that explicitly records over-allocation and under-allocation across locations and shipment categories.

In parallel, broader research has emphasized the digital transformation of transportation and supply chains [20,21], the use of big-data analytics [22], digital twins [23], predictive analytics [24], AI-based risk management [25], data-driven smart manufacturing [26], and machine-learning applications in Industry 4.0 [27]. Deep reinforcement learning has also been examined for inventory control and multi-echelon decision problems [28–30], while resilience-oriented research highlights the management of disruptions and ripple effects [31]. Recent systematic reviews further summarize empirical AI applications in supply chain management [32]. Deep-learning architectures have, for example, been employed to analyze relationships between commodity futures markets and global container-shipping indicators and to improve shipping-market forecasts [33]. These approaches demonstrate

the potential of data-driven logistics systems, but they often focus on prediction rather than transparent shipment-level allocation and may require substantial training data and computational resources.

Against this background, the research gap addressed in this paper is not the isolated use of rules, heuristics, optimization, or historical data. Each of these elements is well established. The unresolved issue is their operational integration into a transparent framework that simultaneously supports:

- (i) The assignment of discrete and non-divisible inbound deliveries;
- (ii) Strict warehouse-specific feasibility and capacity requirements;
- (iii) Short-term balancing of volume and pallet-based utilization;
- (iv) Long-term balancing of shipment categories;
- (v) Corrective adaptation based on realized allocation outcomes across successive planning cycles.

This study therefore proposes a hierarchical two-level allocation framework. The strategic layer applies transparent category-rotation policies that encode long-term operational preferences. The rotation mechanism is not presented as an independently novel scheduling theory; rather, its contribution lies in its formal integration as a policy signal within a constrained allocation framework. The operational layer assigns individual deliveries to feasible warehouse locations by evaluating the normalized marginal effect of each candidate assignment on the global allocation objectives.

The proposed history-driven feedback mechanism differs from conventional rolling-horizon optimization and adaptive metaheuristic penalties in three respects. First, it maintains a signed and normalized imbalance state for every category–location combination, thereby distinguishing historical over-allocation from historical under-allocation. Second, this state is carried across planning cycles and directly modifies the marginal assignment score rather than controlling search operators or merely updating the current input instance. Third, a bounded update rule and a configurable rolling historical window limit overcorrection and reduce oscillatory allocation behavior.

The sequential scoring procedure also differs from a conventional greedy assignment rule. A basic greedy heuristic commonly selects the location with the largest residual capacity or the smallest immediate load increase. In contrast, the proposed method evaluates normalized marginal changes in volume deviation, pallet ratio deviation, category balance, and historical imbalance. The local decision rule is therefore explicitly derived from the components of the global multi-objective formulation and is combined with feasibility filtering, complexity-based delivery ordering, deterministic tie-breaking, and an exception-handling procedure for deliveries without a feasible assignment.

The principal contributions of this study are as follows:

1. A hierarchical two-level decision architecture is developed that separates long-term category-balancing policies from short-term shipment-level allocation decisions.
2. A signed, normalized, and rolling history-driven feedback mechanism is formulated to identify both over-allocation and under-allocation and to correct persistent category–location imbalances across planning cycles.
3. A sequential scoring heuristic is derived from normalized marginal changes in the global objective components, thereby extending conventional greedy assignment with multi-objective, state-dependent, and interpretable decision logic.
4. A reproducible operational evaluation and comparative positioning are provided. The evaluation covers daily simulation instances ranging from 100 to 1500 pallets, with approximately 130 lots per pallet on average, and examines the signed historical mechanism over a rolling-horizon of 30 daily planning periods. In addition, a feature-based comparison with representative optimization, rolling-horizon, heuristic,

data-driven, and simulation-based approaches clarifies the methodological distinction of the proposed framework. Across all evaluated daily instances, the complete allocation procedure was completed in less than 5 s while preserving hard operational constraints.

By integrating policy-based control, mathematical optimization, operational feedback, and interpretable sequential decision-making, the proposed framework seeks to bridge the gap between theoretically rigorous allocation models and practical multi-location inbound logistics operations.

2. Related Work

Research relevant to multi-location inbound allocation spans several interconnected domains, including strategic supply chain design, operational assignment, rolling-horizon optimization, heuristic and hybrid solution methods, adaptive decision mechanisms, and simulation-based warehouse analysis. These streams provide important methodological foundations, but they differ substantially in terms of decision level, allocation unit, temporal treatment, and use of historical information. This section reviews these distinctions and positions the proposed framework relative to existing approaches.

2.1. Strategic and Tactical Allocation Models

Strategic and tactical supply chain models primarily address inventory positioning, network configuration, capacity planning, and aggregate flow allocation. Foundational supply chain and multi-echelon models coordinate inventory positioning, replenishment, and service levels across interconnected facilities [1–4]. Facility location and network design models determine the number, location, and capacity of facilities together with aggregate assignments of customer demand or material flows [5,6]. Advanced planning frameworks subsequently integrate these decisions across procurement, production, distribution, and transportation levels [7,8].

These models provide essential foundations for supply chain coordination but generally operate on aggregated quantities over medium- or long-term planning horizons. Their decision units are commonly inventory quantities, product flows, or customer-demand volumes rather than individual inbound deliveries. Consequently, they do not directly represent shipment indivisibility, sequential assignment decisions, or category-specific operational restrictions encountered in daily inbound logistics.

A related warehouse research stream addresses storage-location assignment, internal product allocation, and layout design. A recent literature review shows that most warehouse allocation studies concentrate on slotting, internal material placement, picking productivity, and layout configuration, while heterogeneous and non-standard operational environments remain comparatively underrepresented [34]. Such research is relevant to warehouse utilization but differs from the present problem, which concerns the allocation of complete inbound deliveries among geographically or operationally distinct warehouse locations before the internal storage decision is made.

2.2. Exact Optimization and Rolling-Horizon Allocation

Mathematical programming enables explicit representation of assignment variables, capacity limits, compatibility restrictions, and multiple operational objectives. Real-time truck-to-door scheduling, for example, has been formulated as a mixed-integer model that assigns inbound vehicles to cross-docking doors while accounting for service and waiting times [14]. Exact optimization provides a valuable benchmark because it can identify globally optimal solutions for tractable instances and quantify the optimality gap of heuristic procedures.

However, exact formulations become increasingly difficult to solve when the number of deliveries, warehouse locations, resource dimensions, and policy constraints grows. Their repeated application may also be problematic in sequential environments in which capacities and balancing states change after every assignment. In such settings, a complete model would need to be re-solved whenever new information becomes available.

Rolling-horizon optimization addresses dynamic information by repeatedly shifting the planning window, updating the current system state, and solving a revised finite-horizon problem. Recent allocation-location research has applied this principle to multi-category emergency supply chains, where allocation and facility decisions are updated across consecutive planning cycles [35]. Rolling-horizon methods therefore provide a systematic mechanism for reacting to changing demand and system conditions.

Nevertheless, the adaptation in a conventional rolling-horizon framework is primarily generated by replacing forecasts, capacities, demands, or other state information before the next optimization run. Historical outcomes affect subsequent decisions through the updated problem instance. By contrast, the mechanism proposed in this study retains a signed category–location imbalance state across planning cycles and introduces that state directly into the marginal evaluation of future shipment assignments. It therefore complements rather than replicates rolling-horizon reoptimization.

2.3. Heuristic and Hybrid Optimization Approaches

Heuristic and metaheuristic methods are widely used when exact optimization becomes computationally prohibitive. Traditional greedy procedures construct a solution sequentially by selecting the locally most attractive feasible option; for example, the warehouse with the greatest remaining capacity or the smallest immediate utilization increase. Such procedures are computationally efficient but may generate myopic decisions because they do not explicitly account for the effect of each assignment on several global balancing objectives.

Adaptive large neighborhood search extends conventional local search by learning which destroy and repair operators are most effective during the search process [15–17]. Hybrid genetic algorithms similarly combine population-based exploration with local improvement and adaptive diversity management [18]. These approaches adapt the internal search strategy and can produce high-quality solutions for difficult combinatorial problems.

Dynamic warehouse task-allocation research has further combined adaptive large neighborhood search with assignment algorithms to allocate tasks among automated guided vehicles under continuously changing system conditions [19]. This stream demonstrates the practical value of adaptive heuristics in warehouse operations. However, its adaptation typically concerns operator selection, routing structures, task queues, or current vehicle states.

The proposed sequential scoring heuristic differs in both purpose and mechanism. It does not adapt search operators and does not conduct an iterative neighborhood search over complete solutions. Instead, it evaluates each feasible shipment–location pair using the normalized marginal change in the components of the global allocation objective. Its decision state includes current resource utilization as well as a persistent historical category–location imbalance. The procedure is therefore closer to a multi-objective constructive heuristic than to a conventional greedy, neighborhood-search, or metaheuristic algorithm.

2.4. Adaptive, Feedback-Driven, and Data-Based Decision Frameworks

Data-driven approaches increasingly support forecasting, inventory control, transportation planning, and warehouse operations. Research spans digital supply chain design, big-data analytics, digital twins, predictive analytics, AI-based risk management, smart manufacturing,

machine learning, and reinforcement-learning-based inventory control [20–32]. Deep-learning methods have, for example, been used to capture nonlinear relationships between commodity futures markets and international container-shipping indicators [33]. Digital supply chain architectures likewise emphasize real-time data integration, virtual representations of operational systems, and continuous decision support [21–23].

These approaches demonstrate how historical and real-time information can improve logistics decisions. However, predictive learning and operational allocation represent distinct methodological tasks. Predictive models estimate future demand, market indicators, or operational states; an allocation procedure must subsequently transform those estimates into feasible shipment-level decisions. Data-intensive models may additionally require substantial training data and can be difficult to interpret in environments where planners must explain and override recommendations.

The historical mechanism proposed in this study is therefore not presented as a predictive machine-learning model. It is an interpretable feedback controller embedded in the allocation objective. Its state variable records the signed difference between realized and target category shares, distinguishes over-allocation from under-allocation, and is updated using a bounded rolling window. This design makes the corrective signal transparent and directly traceable to observed allocation outcomes.

The mechanism also differs from adaptive penalty updates used in metaheuristics. Adaptive penalties typically increase or decrease constraint-violation costs in order to guide a search toward feasible or higher-quality solutions. In the proposed framework, the historical term does not merely regulate search feasibility. It represents an operational imbalance that persists across completed planning cycles and modifies subsequent assignment preferences until the imbalance is corrected.

2.5. Simulation-Based Warehouse Decision Support

Simulation provides a complementary method for analyzing warehouse systems characterized by stochastic arrivals, queues, congestion, interacting resources, and process-dependent service times. Simulation-based optimization has been applied to warehouse worker assignment by combining discrete-event simulation with neighborhood search to balance inbound and outbound service performance under variable workloads [36]. Discrete-event simulation has also been used to evaluate forklift allocation in distribution centers with aisle restrictions and to assess the operational implications of alternative resource configurations [37].

These studies demonstrate that simulation is particularly valuable when system performance depends on queuing, travel, congestion, processing times, or complex resource interactions that are difficult to express analytically. However, simulation alone evaluates the consequences of predefined policies and does not necessarily produce shipment-level assignments unless it is coupled with an optimization or search procedure.

The framework developed in this study focuses on the allocation decision itself and employs an analytical feasibility model together with a constructive heuristic. Simulation is therefore treated as a complementary validation method rather than the primary decision engine. Future extensions may embed the proposed allocation procedure in a discrete-event simulation environment to evaluate downstream effects on dock congestion, handling times, workforce utilization, and internal material flows.

2.6. Comparative Positioning and Research Gap

Table 1 summarizes the principal methodological differences between representative existing approaches and the proposed framework. The comparison distinguishes decision

level, allocation unit, adaptive mechanism, and solution approach because these dimensions are central to identifying the contribution of the present study.

Table 1. Comparison of representative supply chain and warehouse allocation approaches with the proposed framework.

Study	Decision Context and Level	Primary Allocation Unit	Adaptive or Historical Mechanism	Solution Method	Distinction Relative to the Present Study
Snyder and Shen [2]	Strategic and tactical multi-echelon inventory planning	Aggregate inventory quantities	Stochastic inventory-state updates	Analytical and optimization models	Does not address sequential assignment of indivisible inbound deliveries
Melo et al. [5]	Strategic facility location and network design	Aggregate demand and material flows	Generally static or scenario-based	Facility location optimization	Focuses on network configuration rather than daily allocation to existing warehouses
Fatthi et al. [14]	Operational cross-dock assignment	Individual inbound trucks	Current operational state	Mixed-integer programming	Exact operational assignment without persistent category–location feedback
Ropke and Pisinger [17]	Operational routing and pickup-and-delivery planning	Customer and transport requests	Adaptive operator-selection weights	Adaptive large neighborhood search	Historical information guides search operators rather than operational balancing states
Vidal et al. [18]	Operational vehicle routing	Customer visits and routes	Adaptive diversity management	Hybrid genetic algorithm	Adapts population search rather than shipment–location preferences
Xin et al. [19]	Dynamic warehouse task allocation	AGV transport tasks	Current task and vehicle states; adaptive search	ALNS combined with assignment optimization	Focuses on robot-task allocation and routing within a warehouse
Shi et al. [35]	Dynamic allocation–location planning	Multi-category product flows	Rolling-horizon state and demand updates	Repeated optimization	Historical effects enter through updated problem instances rather than a signed imbalance state
Su et al. [33]	Strategic shipping-market analysis	Aggregated market indicators	Learned temporal relationships	Deep learning	Predictive rather than prescriptive shipment-level allocation
Ganbold et al. [36]	Operational warehouse workforce planning	Workers and warehouse activities	Stochastic workload represented in simulation	Simulation-based neighborhood search	Allocates labor resources rather than inbound deliveries
Becerra-Fernandez et al. [37]	Operational distribution center resource planning	Forklifts and handling activities	Scenario-based simulation	Discrete-event simulation	Evaluates equipment configurations rather than generating multi-location shipment assignments
This study	Strategic policy and operational allocation levels	Discrete, non-divisible inbound deliveries	Signed, normalized, bounded, rolling category–location imbalance state	Feasibility filtering and normalized marginal scoring heuristic, with exact MILP benchmark for small instances	Integrates long-term policy signals, short-term feasibility, and persistent corrective feedback in an interpretable shipment-level framework

The comparison indicates that the novelty of the present study does not arise from introducing rules, optimization, heuristic construction, or historical information individually. Rather, it lies in their formal integration around a specific operational decision unit: the complete inbound delivery.

Four gaps are addressed simultaneously. First, deliveries are treated as discrete and non-divisible units rather than aggregate flows. Second, warehouse-specific feasibility and resource capacities are enforced before candidate assignments are scored. Third, the sequential heuristic evaluates normalized marginal changes in global volume, pallet ratio, category-balance, and historical objectives rather than applying a single greedy priority criterion. Fourth, the feedback mechanism retains a signed and bounded category–location imbalance state across planning cycles, allowing the system to distinguish historical over-allocation from under-allocation and to make corrective assignments without relying on a black-box learning model.

The strategic rotation mechanism is not claimed as a new scheduling theory. Its role is to encode transparent operational policy preferences and provide a long-term balancing signal to the operational layer. The methodological contribution is the in-

tegration of this exogenous policy signal with normalized objective-based scoring and endogenous historical correction. This structure enables the framework to preserve feasibility, remain computationally efficient, and provide an explicit explanation for every assignment recommendation.

3. Problem Definition and Hierarchical Two-Level System Architecture

3.1. Operational Setting and Decision Scope

The problem considered in this study concerns the short-term allocation of inbound deliveries across a set of heterogeneous warehouse locations. Let (I) denote the set of inbound deliveries and ($J = \{1, \dots, m\}$) the set of available warehouse locations. The general framework is defined for an arbitrary number of warehouses, while the experimental implementation presented in Section 6 considers six anonymized locations, denoted by W1–W6, with different capacities and handling capabilities.

Each delivery ($i \in I$) represents a discrete operational unit and is characterized by a pallet quantity, a shipment volume, one or more category indicators, and location-dependent handling requirements. Relevant categories may include heavy, hazardous, seasonal, or otherwise operationally sensitive deliveries. Warehouse locations may differ with respect to storage capacity, handling equipment, admissible shipment categories, and operational workload limits.

The delivery is treated as non-divisible. Under normal operating conditions, it must therefore be assigned entirely to one warehouse location. Splitting an inbound delivery across multiple facilities is excluded because it would require additional synchronization, transportation, documentation, and handling activities. This differs from strategic supply chain and inventory models that commonly represent inventory or product flows as divisible and aggregated quantities [1–8].

The model distinguishes between physical feasibility and operational balancing. In the considered application, pallet positions or handling units constitute the binding storage resource and are therefore represented as hard capacity constraints. Shipment volume is retained as a separate operational balancing indicator because it reflects workload, space consumption, and the relative size of inbound flows, even when the formal storage limit is expressed in pallets. This distinction enables the system to prevent physical capacity violations while simultaneously balancing the operational workload among warehouse locations.

The generalized formulation introduced in Section 4 additionally permits multiple resource dimensions, including pallet positions, physical volume, weight, or processing capacity. The experimental setting uses pallet capacity as the principal hard resource, while weight- and category-related restrictions are represented through warehouse-delivery compatibility parameters. This modeling choice reflects the operational setting without limiting the general applicability of the framework.

The objective is to determine warehouse assignments that satisfy all mandatory operational requirements while minimizing deviations from desired utilization, proportional allocation, category distribution, and historical balancing targets. The problem is therefore positioned between strategic network design and internal warehouse slotting: the warehouse network is assumed to exist, but complete inbound deliveries must be assigned to one of its locations during each operational planning cycle.

In the operational implementation, the allocation procedure is executed once per day. The resulting warehouse assignments are generated for the subsequent operational day rather than being continuously recalculated during ongoing warehouse operations.

3.2. Hard Feasibility Requirements and Soft Operational Objectives

The allocation problem incorporates two conceptually distinct groups of requirements: hard feasibility constraints and soft operational objectives.

Hard constraints define the admissible solution space and cannot be violated by an automatically accepted assignment. First, the cumulative resource consumption of deliveries assigned to a warehouse must remain within its available capacity. Second, each delivery may only be assigned to warehouse locations that possess the required infrastructure, equipment, regulatory authorization, or handling capability. Third, a delivery cannot be divided among several warehouse locations within the same planning cycle.

If no feasible warehouse exists for a delivery, the delivery is not treated as physically discarded or permanently rejected. Instead, it is transferred to an operational exception queue for manual intervention, rescheduling, capacity adjustment, or delayed assignment. This exception state is explicitly represented in the revised mathematical model through an auxiliary decision variable and is associated with a large penalty so that it is used only when no regular feasible assignment exists.

Within the feasible solution space, the framework evaluates several soft objectives. The first objective concerns shipment volume balancing. It measures the deviation between the allocated volume and the target volume of each warehouse location. The second objective concerns pallet-based proportional allocation and evaluates whether the assigned pallet quantities are consistent with predefined warehouse shares. The use of both measures is intentional: pallet quantity represents the principal physical capacity resource, whereas shipment volume provides a complementary workload-balancing indicator.

The third objective concerns category-specific distribution. Operationally sensitive shipment categories should not be systematically concentrated at a single warehouse unless this concentration is required by feasibility restrictions or explicit management policy. The fourth objective concerns historical correction. Realized allocations from previous planning cycles are compared with desired category–location shares, and persistent over-allocation or under-allocation is incorporated into subsequent decisions.

All soft-objective components are normalized before aggregation. This prevents differences in physical units and numerical magnitudes from causing one objective to dominate the scoring function solely because it is measured on a larger scale.

3.3. Hierarchical Two-Level System Architecture

The proposed framework is organized as a hierarchical two-level decision system that separates longer-term policy guidance from short-term operational allocation. Figure 1 presents the overall architecture and illustrates the interaction among input processing, strategic rotation control, operational allocation, decision support, and historical feedback.

The first level represents the strategic policy and rotation layer. It generates category–location preferences over a longer sequence of planning periods. The second level represents the operational allocation layer. It assigns individual deliveries under current feasibility, capacity, and balancing conditions. Historical allocation outcomes are subsequently returned to the framework and used to adjust future preferences and targets.

The architecture processes three principal information groups. Delivery data describe pallet quantity, volume, category membership, and handling requirements. Warehouse data describe resource capacities and compatibility conditions. Historical data contain realized category–location allocations and the corresponding deviations from target shares.

After validation, preprocessing, and aggregation, the strategic layer produces a policy signal for the operational allocation engine. The operational engine then filters infeasible warehouse options and evaluates the normalized marginal impact of each feasible assignment. The resulting recommendations are presented through the decision-support

component, where authorized planners may review exceptional cases and document manual adjustments.

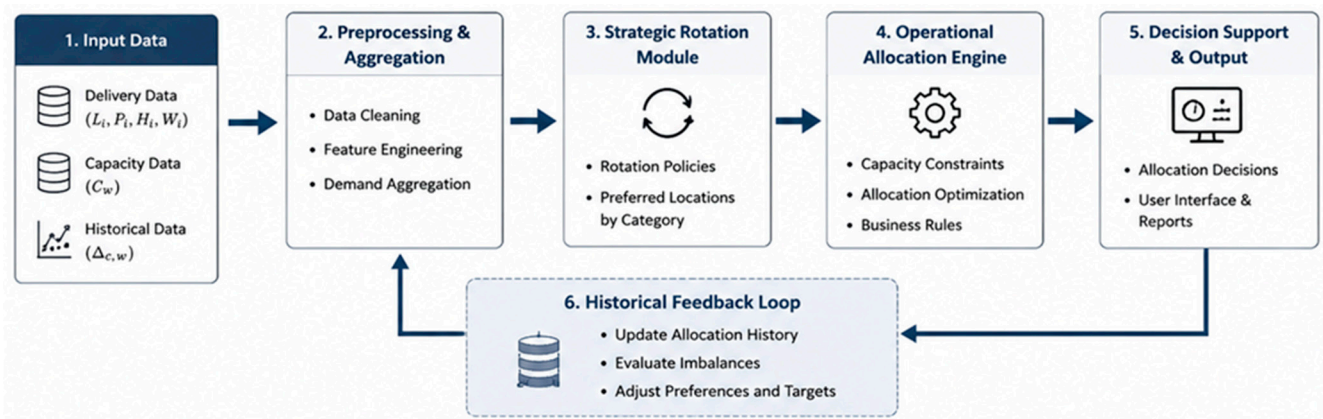


Figure 1. Hierarchical two-level architecture of the proposed inbound allocation framework, integrating operational input data, preprocessing, strategic category-rotation policies, shipment-level allocation, decision support, and rolling historical feedback.

3.3.1. Strategic Policy and Rotation Layer

The strategic layer provides long-term category-balancing guidance. Its purpose is not to determine the final warehouse assignment independently, but to generate a transparent preference signal that is considered by the operational allocation engine.

The rotation policy may be defined as a cyclic preference sequence, a target category share, or a warehouse-priority order specified by logistics management. For example, a category may preferably alternate among suitable warehouses in order to avoid persistent concentration, while another category may follow an unequal rotation because one location has substantially greater capacity or superior handling capability.

The practical basis of the mechanism is therefore an explicit operational policy derived from warehouse capacities, infrastructure conditions, regulatory requirements, and management preferences. It is not presented as a new standalone scheduling theory. Its methodological role lies in formalizing these policy preferences and integrating them into a shipment-level optimization and feedback framework.

The rotation signal is treated as a soft preference rather than a mandatory assignment rule. Hard feasibility constraints always take precedence. If the preferred warehouse lacks sufficient capacity or is incompatible with the current delivery, the operational layer evaluates the remaining feasible locations. This prevents the strategic policy from producing physically infeasible decisions.

3.3.2. Operational Allocation Layer

The operational layer assigns individual inbound deliveries within each daily planning cycle. Before the allocation procedure begins, all deliveries available for the subsequent operational day are collected and ordered according to their operational complexity. This ordering gives priority to highly constrained or resource-intensive deliveries. The quantitative ordering rule is defined in Section 5 and considers the number of feasible warehouse locations, restrictive category indicators, pallet quantity, and shipment volume.

The complete allocation procedure is executed once per day for the full set of available deliveries. Within this daily batch-oriented run, however, the deliveries are processed sequentially. For each delivery, the framework constructs the currently feasible warehouse set by evaluating the remaining resource capacities and the relevant delivery–warehouse compatibility conditions.

Each feasible candidate assignment is subsequently evaluated according to its normalized marginal effect on the allocation objectives. These effects include changes in shipment volume deviation, pallet ratio deviation, category distribution deviation, strategic rotation adherence, and projected historical imbalance.

This procedure differs from a conventional greedy rule that selects the warehouse with the greatest residual capacity or the smallest immediate load increase. The proposed operational layer evaluates several current and historical consequences simultaneously and directly derives each local score from the objective components of the global mathematical formulation.

Following each assignment, the remaining warehouse capacities, allocated volumes, pallet quantities, current-day category quantities, and projected historical imbalance are updated. The revised internal state is then used immediately when evaluating the next delivery in the ordered daily batch. Consequently, every assignment reflects all preceding decisions made during the same daily planning run.

After all deliveries have been assigned or transferred to the exception procedure, the realized daily category–warehouse deviations are calculated and added once to the rolling 30-day historical record. This updated historical record is then used to determine the historical state for the subsequent planning day. The tie-breaking rules, exception-handling procedure, sequential state updates, and complete algorithmic sequence are specified in Section 5.

3.4. Signed Rolling Historical Feedback Mechanism

The historical feedback mechanism records whether a category has been allocated above or below its intended share at each warehouse location. For every category–location combination, the framework maintains a signed and normalized imbalance state. A positive value represents historical over-allocation, whereas a negative value represents historical under-allocation.

This signed representation is essential because a non-negative historical penalty would only discourage additional assignments without identifying the direction in which correction is required. Under the revised formulation, an assignment to an historically over-allocated warehouse increases the score, whereas an assignment to an historically under-allocated warehouse may reduce the historical component of the score.

The imbalance state is calculated over a configurable rolling window covering the most recent planning cycles. Older observations may receive lower weights through a decay factor, allowing the mechanism to remain responsive to current conditions while retaining relevant historical information. The state is normalized by the total category quantity observed during the window and bounded within a predefined interval.

A damping parameter limits the magnitude of the update between consecutive planning cycles. This prevents a single atypical planning period from reversing allocation preferences excessively and reduces the risk of overcorrection or oscillatory behavior. The formal update equations, normalization procedure, rolling-window definition, and stability interpretation are presented in Section 4.

The feedback mechanism differs from rolling-horizon optimization because it does not merely re-solve the same allocation model with an updated demand instance. It retains a persistent corrective category–location state across completed planning cycles. It also differs from adaptive penalty mechanisms in metaheuristics because the feedback value represents an observed operational imbalance rather than the performance of a search operator or the frequency of constraint violations.

In the evaluated implementation, the historical feedback state is based on a rolling window containing the most recent 30 completed daily planning periods. After each daily

allocation run, the newly realized category–warehouse deviations are added to the historical record, while observations older than 30 days are removed. During system initialization, when fewer than 30 completed periods are available, all available observations are used and the historical weights are renormalized over the effective window length.

Figure 2 summarizes the sequential workflow performed within the daily batch-oriented allocation run. All deliveries available for the subsequent operational day are collected before the procedure starts. The algorithm then processes these deliveries sequentially, updates the operational state and projected historical imbalance after each assignment, and uses the revised state when evaluating the next delivery. After the complete delivery batch has been processed, the realized daily deviations are added to the rolling 30-day historical record.

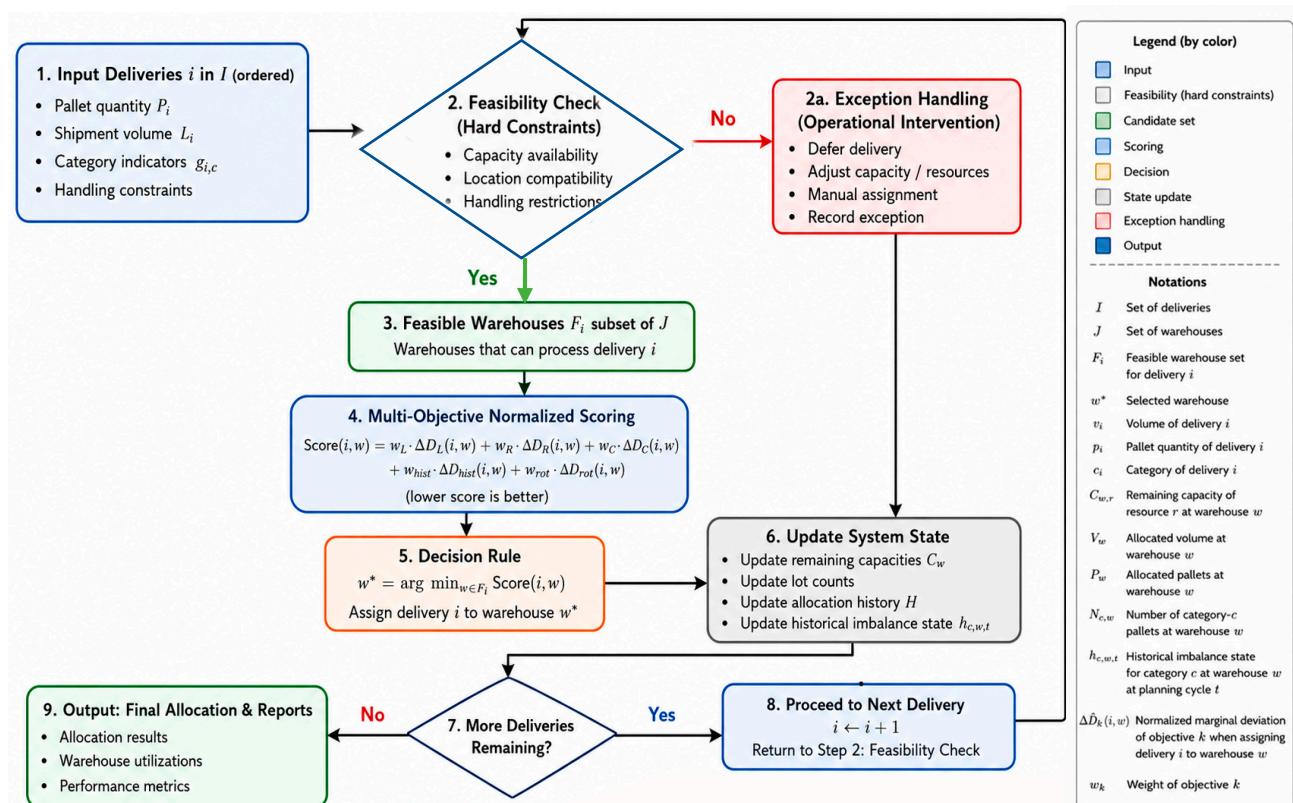


Figure 2. Sequential decision workflow of the operational allocation engine, including feasibility filtering, normalized multi-objective marginal scoring, exception handling, state updating, and transition to the next inbound delivery. All mathematical symbols are defined in Table 2 and Sections 4 and 5.

The integration of the two decision levels enables the framework to combine explicit operational policy, short-term feasibility, and long-term corrective behavior. The resulting decisions remain interpretable because every recommendation can be decomposed into its feasibility conditions, policy preference, current balancing effects, and historical feedback contribution.

4. Mathematical Model

4.1. Model Scope and Assumptions

The multi-location inbound allocation problem is formulated as a constrained multi-objective binary assignment model. The model is solved for each operational planning cycle (t), during which a set of discrete inbound deliveries must be assigned to a set of warehouse locations.

Each delivery represents a non-divisible operational unit. Under normal operating conditions, it must therefore be assigned entirely to one warehouse. Partial assignment across multiple facilities is excluded because it would generate additional transportation, synchronization, documentation, and handling requirements.

The formulation distinguishes between hard feasibility constraints and soft allocation objectives. Hard constraints represent warehouse resource capacities and shipment–location compatibility and cannot be violated by an automatically accepted assignment. Soft objectives represent operational preferences concerning volume balance, pallet-distribution ratios, category allocation, strategic rotation, and historical correction.

The generalized formulation supports multiple hard capacity dimensions, such as pallet positions, physical volume, weight, processing capacity, or dock-handling capacity. In the experimental application, pallet positions constitute the principal hard storage resource. Shipment volume is additionally considered as a soft balancing dimension because it represents the relative magnitude and operational workload of inbound flows. Weight-related and category-dependent restrictions are represented through compatibility parameters where they do not constitute independent numeric capacity dimensions.

If no feasible warehouse exists for a delivery, the delivery is transferred to an exception state for operational intervention, delayed assignment, or capacity adjustment. This state is formally represented by an auxiliary binary variable and is assigned a sufficiently large penalty so that it is selected only when no regular feasible allocation is available.

The complete notation used in the mathematical formulation is summarized in Table 2.

Table 2. Comprehensive notation used in the multi-location inbound allocation model.

Symbol	Type	Definition and Unit
(I_t)	Set	Inbound deliveries available in planning cycle (t)
(J)	Set	Warehouse locations
(C)	Set	Shipment categories
(R)	Set	Hard capacity-resource dimensions
(i)	Index	Delivery, $(i \in I_t)$
(w)	Index	Warehouse location, $(w \in J)$
(c)	Index	Shipment category, $(c \in C)$
(r)	Index	Capacity resource, $(r \in R)$
(t)	Index	Operational planning cycle
(L_i)	Parameter	Shipment volume of delivery (i)
(P_i)	Parameter	Pallet quantity of delivery (i)
$(q_{i,r})$	Parameter	Consumption of resource (r) by delivery (i)
$(C_{w,r,t})$	Parameter	Available capacity of resource (r) at warehouse (w) in cycle (t)
$(g_{i,c})$	Binary parameter	Equals 1 if delivery (i) belongs to category (c)
$(b_{i,c})$	Parameter	Category-balancing quantity of delivery (i) for category (c)
$(a_{i,w})$	Binary parameter	Equals 1 if delivery (i) is compatible with warehouse (w)
$(\alpha_{w,t})$	Parameter	Target share of total shipment volume assigned to warehouse (w)
$(\beta_{w,t})$	Parameter	Target share of total pallet quantity assigned to warehouse (w)
$(\gamma_{c,w})$	Parameter	Target share of category (c) assigned to warehouse (w)
$(\pi_{c,w,t})$	Parameter	Strategic rotation-preference value for category (c) and warehouse (w)

Table 2. Cont.

Symbol	Type	Definition and Unit
(H)	Parameter	Length of the rolling historical window; $H = 30$ daily planning periods in the experimental implementation
(ρ)	Parameter	Historical decay parameter
(η)	Parameter	Historical correction gain
($\bar{h}$)	Parameter	Maximum absolute historical imbalance
($h_{c,w,t}$)	Parameter	Signed historical imbalance state at the start of cycle (t)
($x_{i,w}$)	Binary variable	Equals 1 if delivery (i) is assigned to warehouse (w)
(y_i)	Binary variable	Equals 1 if delivery (i) enters the operational exception procedure
($d^{k,+}, d^{k,-}$)	Continuous variables	Positive and negative deviation variables used for linearization
($w_L, w_R, w_C, w_{\text{hist}}, w_{\text{rot}}$)	Parameters	Weights of the normalized soft-objective components
(M_{exc})	Parameter	Dominating penalty assigned to an exception decision

The category-balancing quantity ($b_{i,c}$) allows different operational interpretations. If category balance is evaluated by delivery count, then ($b_{i,c} = g_{i,c}$). If it is evaluated according to pallet quantity, then ($b_{i,c} = P_i g_{i,c}$). The selected definition must remain consistent throughout the experimental evaluation.

4.2. Decision Variables and Complete Assignment

The regular assignment decision is represented by the binary variable:

$$(x_{i,w} \in 0, 1 \forall i \in I_t, \forall w \in J) \quad (1)$$

where ($x_{i,w} = 1$) indicates that delivery (i) is assigned to warehouse (w).

The exception variable is defined as:

$$(y_i \in 0, 1 \forall i \in I_t) \quad (2)$$

where ($y_i = 1$) indicates that delivery (i) cannot be assigned through the regular automated procedure and must be deferred or transferred to operational intervention.

Each delivery must either be assigned entirely to one warehouse or transferred to the exception procedure:

$$\left(\sum_{w \in J} x_{i,w} + y_i = 1 \forall i \in I_t \right) \quad (3)$$

Equation (3) preserves shipment non-divisibility while making the mathematical formulation consistent with the operational exception path shown in Figure 2.

4.3. Multi-Resource Capacity and Compatibility Constraints

The generalized formulation permits several hard capacity dimensions. For each warehouse (w), resource consumption may not exceed the corresponding available capacity:

$$\left(\sum_{i \in I_t} q_{i,r} x_{i,w} \leq C_{w,r,t} \forall w \in J, \forall r \in R \right) \quad (4)$$

For the pallet capacity dimension ($r = p$), the resource consumption is defined as:

$$(q_{i,p} = P_i) \quad (5)$$

If physical volume, weight, processing capacity, or another resource is also modeled as a hard limitation, the corresponding resource consumption and warehouse capacity are included analogously.

In the experimental application, pallet positions constitute the binding hard capacity resource. Shipment volume is retained as a soft balancing objective because two deliveries with similar pallet quantities may still differ substantially in physical size and handling workload.

Warehouse-delivery compatibility is enforced through:

$$(x_{i,w} \leq a_{i,w} \forall i \in I_t, \forall w \in J) \quad (6)$$

The parameter $(a_{i,w})$ incorporates handling restrictions, infrastructure requirements, safety regulations, category limitations, and other operational conditions. An incompatible warehouse can therefore never be selected by the automated allocation procedure.

4.4. Dynamic Volume and Pallet Targets

The total inbound shipment volume and total pallet quantity in planning cycle (t) are defined as:

$$(V_t = \sum_{i \in I_t} L_i) \quad (7)$$

$$(P_t = \sum_{i \in I_t} P_i) \quad (8)$$

The model assigns a target volume share $(\alpha_{w,t})$ and a target pallet share $(\beta_{w,t})$ to each warehouse:

$$(\sum_{w \in J} \alpha_{w,t} = 1) \quad (9)$$

$$(\sum_{w \in J} \beta_{w,t} = 1) \quad (10)$$

Under nominal demand conditions, the volume shares correspond to predefined operational targets:

$$(\alpha_{w,t} = \bar{\alpha}_w \text{ for } V^{min} \leq V_t \leq V^{max}) \quad (11)$$

Under low-load or high-load conditions, the volume shares are adjusted according to the currently available principal capacity:

$$(\alpha_{w,t} = \frac{C_{w,p,t}}{\sum_{u \in J} C_{u,p,t}} \text{ for } V_t < V^{min} \text{ or } V_t > V^{max}) \quad (12)$$

The resulting warehouse targets are:

$$(T_{w,t}^L = \alpha_{w,t} V_t \forall w \in J) \quad (13)$$

$$(T_{w,t}^P = \beta_{w,t} P_t \forall w \in J) \quad (14)$$

This formulation enables the target structure to respond to changes in total inbound demand and warehouse resource availability.

4.5. Normalized Immediate Allocation Deviations

Because the objective components are expressed in different physical units, all deviations are normalized before aggregation.

The normalized shipment volume deviation is defined as:

$$(\widehat{D}_L = \frac{1}{\max(V_t, \varepsilon)} \sum_{w \in J} \left| \sum_{i \in I_t} L_i x_{i,w} - T_{w,t}^L \right|) \quad (15)$$

where ($\varepsilon > 0$) prevents division by zero.

The normalized pallet ratio deviation is:

$$(\widehat{D}_R = \frac{1}{\max(P_t, \varepsilon)} \sum_{w \in J} \left| \sum_{i \in I_t} P_i x_{i,w} - T_{w,t}^P \right|) \quad (16)$$

For each active category (c), the total category-balancing quantity is:

$$(B_{c,t} = \sum_{i \in I_t} b_{i,c}) \quad (17)$$

The set of active categories is defined as:

$$(C_t = c \in C : B_{c,t} > 0) \quad (18)$$

The normalized category distribution deviation is:

$$\left(\widehat{D}_C = \frac{1}{|C_t|} \sum_{c \in C_t} \frac{1}{B_{c,t}} \sum_{w \in J} \left| \sum_{i \in I_t} b_{i,c} x_{i,w} - \gamma_{c,w} B_{c,t} \right| \right) \quad (19)$$

The target category shares satisfy:

$$\left(\sum_{w \in J} \gamma_{c,w} = 1 \quad \forall c \in C \right) \quad (20)$$

Equations (15)–(19) yield dimensionless values and prevent one objective from dominating solely because it is measured on a numerically larger scale.

4.6. Strategic Rotation Preference

The strategic policy layer generates a preference parameter ($\pi_{c,w,t} \in [0, 1]$), where a larger value indicates a stronger preference for assigning category (c) to warehouse (w) during cycle (t).

The normalized rotation policy deviation is defined as:

$$\left(\widehat{D}_{\text{rot}} = \frac{\sum_{i \in I_t} \sum_{c \in C_t} \sum_{w \in J} b_{i,c} (1 - \pi_{c,w,t}) x_{i,w}}{\max(\sum_{c \in C_t} B_{c,t}, \varepsilon)} \right) \quad (21)$$

The rotation component is a soft objective. It cannot override capacity or compatibility constraints. Its role is to translate transparent operational policies into a preference signal considered by the allocation model.

4.7. Signed Rolling Historical Imbalance

The historical feedback mechanism must distinguish over-allocation from under-allocation. For each category (c), warehouse (w), and previous planning cycle (τ), the realized allocation share is:

$$\left(\phi_{c,w,\tau} = \frac{\sum_{i \in I_\tau} b_{i,c} x_{i,w,\tau}}{B_{c,\tau}} \text{ if } B_{c,\tau} > 0 \right) \quad (22)$$

The corresponding signed deviation is:

$$(\delta_{c,w,\tau} = \phi_{c,w,\tau} - \gamma_{c,w}) \quad (23)$$

A positive value of ($\delta_{c,w,\tau}$) indicates historical over-allocation, whereas a negative value indicates historical under-allocation.

The rolling historical weights are defined by:

$$\left(\omega_k = \frac{(1-\rho)\rho^{k-1}}{1-\rho^H} \quad k = 1, \dots, H, \quad 0 < \rho < 1 \right) \quad (24)$$

where ($k = 1$) represents the most recent completed planning cycle. The weights satisfy:

$$\left(\sum_{k=1}^H \omega_k = 1 \right) \quad (25)$$

The historical imbalance state available at the beginning of cycle (t) is:

$$h_{c,w,t} = \min \left\{ \bar{h}, \max \left\{ -\bar{h}, \sum_{k=1}^H \omega_k \delta_{c,w,t-k} \right\} \right\} \quad (26)$$

For the experimental implementation, the rolling-window length is fixed at $H = 30$ daily planning periods. During the initialization phase, the effective window length is $H_t = \min H, t - 1$. The historical weights are renormalized over the H_t observations available at the beginning of planning cycle t .

Where:

$$\text{clip}(z, -\bar{h}, \bar{h}) = \min(\bar{h}, \max(-\bar{h}, z)) \quad (27)$$

The projected imbalance after including the current allocation decision is:

$$\left(\widetilde{h_{c,w,t}}(x) = h_{c,w,t} + \eta \left(\frac{\sum_{i \in I_t} b_{i,c} x_{i,w}}{B_{c,t}} - \gamma_{c,w} \right) \right) \quad (28)$$

where ($0 < \eta \leq 1$) controls the strength of the current correction.

The normalized historical deviation is then:

$$\left(D_{\text{hist}} = \frac{1}{|C_t| |J|} \sum_{c \in C_t} \sum_{w \in J} \left| \widetilde{h_{c,w,t}}(x) \right| \right) \quad (29)$$

This formulation produces directional correction. If ($h_{c,w,t} > 0$), the warehouse has historically received more than its target share of category (c), and further assignments increase the historical penalty. If ($h_{c,w,t} < 0$), an additional assignment can move the projected imbalance toward zero and reduce the penalty.

The mechanism is bounded because ($h_{c,w,t} \in [-\bar{h}, \bar{h}]$), the historical weights sum to one, and the correction gain satisfies ($0 < \eta \leq 1$). These conditions limit overcorrection. The rolling window and decay parameter reduce the influence of old observations, while the clipping bound prevents individual periods from generating unbounded feedback. Although the discrete allocation process does not permit a general asymptotic stability proof, the historical state itself remains bounded and the selected parameters reduce oscillatory behavior.

At the end of cycle (t), the realized deviation ($\delta_{c,w,t}$) is added to the historical window, the oldest observation is removed, and Equation (26) is recomputed for cycle ($t + 1$).

4.8. Objective Function and Weight Selection

Because all soft-objective components are normalized, the weighting parameters represent relative operational priorities rather than corrections for different physical units. The weights were specified before the final experimental evaluation based on the operational importance of volume balance, pallet ratio consistency, category distribution, historical correction, and strategic rotation adherence. The final weight vector was fixed for all evaluated scenarios, benchmark methods, and repetitions and was not adjusted using the

reported test results. The robustness of the selected historical feedback weight and the associated historical parameters is examined in Section 6.7.

The complete objective function is:

$$\left(\min Z = M_{\text{exc}} \sum_{i \in I_t} y_i + w_L \widehat{D}_L + w_R \widehat{D}_R + w_C \widehat{D}_C + w_{\text{hist}} \widehat{D}_{\text{hist}} + w_{\text{rot}} \widehat{D}_{\text{rot}} \right) \quad (30)$$

The exception penalty (M_{exc}) is selected so that avoiding an exception dominates every possible improvement in the soft-objective components. Consequently, the model assigns a delivery to the exception procedure only when no complete feasible assignment exists or when the complete assignment of all deliveries is impossible under the available capacities.

The normalized soft-objective weights satisfy:

$$(w_L + w_R + w_C + w_{\text{hist}} + w_{\text{rot}} = 1) \quad (31)$$

$$(w_L, w_R, w_C, w_{\text{hist}}, w_{\text{rot}} \geq 0) \quad (32)$$

The normalized objective components, their operational interpretations, and the corresponding weighting parameters are summarized in Table 3.

Table 3. Components of the normalized multi-objective function and corresponding weighting parameters.

Objective Component	Operational Interpretation	Weight
$(\widehat{D}_L)$	Normalized shipment volume deviation	(w_L)
$(\widehat{D}_R)$	Normalized pallet ratio deviation	(w_R)
$(\widehat{D}_C)$	Normalized category distribution deviation	(w_C)
$(\widehat{D}_{\text{hist}})$	Signed rolling historical correction	(w_{hist})
$(\widehat{D}_{\text{rot}})$	Strategic rotation policy deviation	(w_{rot})
$(\sum_i y_i)$	Number of deliveries transferred to exception handling	(M_{exc})

4.9. Linearization of Absolute Deviation Terms

The objective contains absolute-value expressions. To solve the formulation as a mixed-integer linear program, each absolute term is replaced by positive and negative deviation variables.

For a generic expression (e), introduce $(d^+ \geq 0)$ and $(d^- \geq 0)$ such that:

$$(e = d^+ - d^-) \quad (33)$$

The corresponding absolute value is represented by:

$$(|e| = d^+ + d^-) \quad (34)$$

For example, the warehouse-volume deviation is linearized as:

$$\left(\sum_{i \in I_t} L_i x_{i,w} - T_{w,t}^L = d^{L,+} w - d^{L,-} w \quad \forall w \in J \right) \quad (35)$$

The absolute deviation in Equation (15) is then replaced by:

$$(d^{L,+}w + d^{L,-}w) \quad (36)$$

Similarly, the historical expression is linearized through:

$$\left(h_{c,w,t} + \eta \left(\frac{\sum_{i \in I_t} b_{i,c} x_{i,w}}{B_{c,t}} - \gamma_{c,w} \right)\right) = d^{H,+}c,w - d^{H,-}c,w \quad (37)$$

and:

$$\left(\widetilde{h_{c,w,t}(x)}\right) = d^{H,+}c,w + d^{H,-}c,w \quad (38)$$

The pallet ratio and category distribution expressions are linearized in the same manner. Because all denominators are constants within a planning cycle, the resulting formulation remains linear. Together with binary variables $(x_{i,w})$ and (y_i) , the complete model is therefore a mixed-integer linear program.

4.10. Relationship Between the Global Model and Sequential Scoring

The sequential heuristic introduced in Section 5 is directly derived from the normalized global objective.

Let (S) denote the current partial allocation state and let $(S \oplus (i, w))$ denote the state obtained by assigning delivery (i) to warehouse (w) . The marginal score is defined as:

$$\text{Score}(i, w) = Z_{\text{soft}}(S \oplus (i, w)) - Z_{\text{soft}}(S) \quad (39)$$

where:

$$(Z_{\text{soft}} = w_L \widehat{D}_L + w_R \widehat{D}_R + w_C \widehat{D}_C + w_{\text{hist}} \widehat{D}_{\text{hist}} + w_{\text{rot}} \widehat{D}_{\text{rot}}) \quad (40)$$

Thus, the heuristic does not employ an unrelated local rule. It selects the feasible warehouse that produces the smallest normalized marginal increase—or largest reduction—in the same objective components used by the global mathematical formulation.

The exact MILP optimizes all assignments simultaneously, whereas the heuristic minimizes the marginal objective change sequentially. This relationship enables direct comparison between both methods on small instances and allows the optimality gap of the heuristic to be quantified in the experimental evaluation.

4.11. Model Interpretation

The revised formulation integrates shipment-level feasibility, multiple resource capacities, dynamic allocation targets, strategic policy preferences, and signed historical correction within a unified model.

The distinction between pallet capacity and shipment volume balancing is explicit. Pallet quantity represents a binding physical capacity dimension in the experimental setting, while shipment volume represents a complementary workload-balancing objective. Additional capacity dimensions can be activated through the resource index (r) without altering the general model structure.

The historical term no longer treats imbalance as a non-negative quantity. Instead, it records the direction and magnitude of historical deviation and therefore supports corrective assignment toward under-allocated warehouses while discouraging continued allocation to over-allocated locations.

The exception variable aligns the mathematical formulation with the operational workflow shown in Figure 2. Finally, normalization and absolute-value linearization make the weighting parameters interpretable and enable the exact formulation to be solved as an MILP for benchmark instances.

5. Solution Approach

5.1. Solution Strategy

The mathematical formulation introduced in Section 4 can be solved as a mixed-integer linear program for problem instances of tractable size. Exact optimization is therefore retained as a reference method for small-scale instances and is used in Section 6 to assess the solution quality of the proposed operational procedure.

For larger instances, however, repeated exact optimization may become difficult to apply within short operational decision windows. The number of binary assignment variables increases with the number of deliveries and warehouse locations, while the inclusion of multiple capacity dimensions, compatibility restrictions, category-balancing targets, and historical feedback terms further enlarges the search space.

To support operational-scale allocation, this study therefore employs a constructive sequential heuristic based on normalized marginal objective changes. The heuristic does not use an independent or arbitrary local priority rule. Instead, it evaluates the effect of each candidate assignment on the same normalized objective components defined in the global mathematical model.

The proposed procedure differs from a conventional greedy allocation rule in three principal respects. First, it evaluates multiple operational objectives simultaneously rather than selecting a warehouse only according to residual capacity or immediate utilization. Second, it incorporates the signed historical imbalance state and the strategic rotation preference into each assignment decision. Third, it applies explicit feasibility filtering, complexity-based delivery ordering, deterministic tie-breaking, and formal exception handling.

Hard capacity and compatibility constraints are evaluated before candidate assignments are scored. Consequently, infeasible warehouse options are excluded and cannot be selected through compensation by improvements in soft-objective components.

Operationally, the procedure is executed as a daily batch-oriented planning process. Each run processes the available inbound information and generates the warehouse allocation plan for the following operational day. The framework therefore supports rapid next-day planning rather than continuous real-time reassignment.

5.2. Inputs, Outputs, and Initialization

For each planning cycle t , the algorithm receives the delivery set I_t , warehouse set J , resource set R , and category set C .

The delivery-level inputs comprise shipment volume L_i , pallet quantity P_i , resource consumption $q_{i,r}$, category indicators $g_{i,c}$, category-balancing quantities $b_{i,c}$, and warehouse-compatibility parameters $a_{i,w}$.

The warehouse-level inputs include the available resource capacities $C_{w,r,t}$, target volume shares $\alpha_{w,t}$, target pallet shares $\beta_{w,t}$, and category–location target shares $\gamma_{c,w}$. The algorithm further receives the strategic rotation preferences $\pi_{c,w,t}$, the signed historical imbalance states $h_{c,w,t}$, the normalized objective weights, and the numerical tie tolerance ε_{tie} .

The algorithm returns the regular assignment decisions $x_{i,w}$, the exception indicators y_i , final warehouse resource states, realized allocation metrics, historical observations for the next planning cycle, and computational runtime.

At the beginning of the planning cycle, the remaining capacity of each resource is initialized as:

$$C_{w,r}^{\text{rem}} = C_{w,r,t} \quad \forall w \in J, \forall r \in R \quad (41)$$

The cumulative shipment volume, pallet quantity, and category-balancing quantities are initialized as:

$$V_w = 0, P_w = 0, B_{c,w} = 0 \quad \forall w \in J, \forall c \in C \quad (42)$$

The historical state derived from the preceding 30 completed daily periods is fixed at the beginning of the daily allocation run. During the sequential processing of the complete delivery batch, each assignment updates the current-day category quantities and the projected historical imbalance used for evaluating the remaining deliveries. After all deliveries have been assigned or transferred to exception handling, the realized daily deviations are incorporated into the rolling historical record and the historical state for the subsequent planning day is calculated.

5.3. Complexity-Based Delivery Ordering

The order in which deliveries are processed influences the feasibility and quality of a sequential allocation procedure. Highly constrained deliveries should therefore be considered before deliveries that can be processed by many warehouse locations.

The initial feasible warehouse set of delivery i is defined as:

$$F_i^0 = \{w \in J \mid a_{i,w} = 1 \wedge q_{i,r} \leq C_{w,r,t} \forall r \in R\} \quad (43)$$

A dimensionless operational-complexity index is calculated for each delivery:

$$\kappa_i = \lambda_F \left(1 - \frac{|F_i^0|}{|J|} \right) + \lambda_G \frac{\sum_{c \in C} g_{i,c}}{\max(|C|, 1)} + \lambda_P \frac{P_i}{\max\left(\max_{j \in I_t} P_j, \varepsilon\right)} + \lambda_L \frac{L_i}{\max\left(\max_{j \in I_t} L_j, \varepsilon\right)} \quad (44)$$

The ordering weights satisfy:

$$\lambda_F + \lambda_G + \lambda_P + \lambda_L = 1 \quad (45)$$

$$\lambda_F, \lambda_G, \lambda_P, \lambda_L \geq 0 \quad (46)$$

The first term gives higher priority to deliveries with fewer feasible warehouse alternatives. The second term represents the relative number of active category attributes. The third and fourth terms represent normalized pallet quantity and shipment volume.

Deliveries are processed in descending order of κ_i . If two deliveries have identical complexity-index values, the following deterministic secondary criteria are applied in sequence:

1. Fewer initially feasible warehouse locations;
2. Larger pallet quantity;
3. Larger shipment volume; and
4. Lower original delivery sequence index.

This ordering reduces the probability that scarce warehouse capacity is consumed by flexible deliveries before more restrictive deliveries have been processed.

The ordering weights were specified before the final experimental evaluation to prioritize deliveries with limited warehouse alternatives and high resource requirements. The selected values remain fixed across all scenarios, benchmark methods, and repetitions.

5.4. Dynamic Feasibility Filtering

Let S_k denote the current allocation state before the k -th delivery is processed. The dynamically feasible warehouse set is defined as:

$$F_i(S_k) = \{w \in J \mid a_{i,w} = 1 \wedge q_{i,r} \leq C_{w,r}^{\text{rem}} \forall r \in R\} \quad (47)$$

Only warehouse locations contained in $F_i(S_k)$ are evaluated by the scoring function.

If the feasible warehouse set is empty:

$$F_i(S_k) = \emptyset \quad (48)$$

the delivery is transferred to the operational exception procedure:

$$y_i = 1, x_{i,w} = 0 \forall w \in J \quad (49)$$

The standard automatic procedure does not relax hard capacity or compatibility constraints. It also does not apply unrestricted backtracking because such a mechanism would reduce runtime predictability and make the allocation sequence more difficult to explain.

A delivery entering the exception procedure is not physically rejected or removed from the logistics process. It is recorded in an exception queue and may subsequently be deferred, manually assigned, processed after a capacity adjustment, or included in a separate reoptimization procedure.

This treatment makes the algorithm consistent with the exception variable introduced in Equation (3) and with the operational workflow shown in Figure 2.

5.5. Normalized Marginal Objective Evaluation

For each feasible warehouse $w \in F_i(S_k)$, the algorithm constructs the projected allocation state $S_k \oplus (i, w)$, representing the state resulting from assigning delivery i to warehouse w .

For each normalized objective component $z \in L, R, C, hist, rot$, the marginal objective change is defined as:

$$\Delta \widehat{D}_z(i, w) = \widehat{D}_z(S_k \oplus (i, w)) - \widehat{D}_z(S_k) \quad (50)$$

The complete candidate score is calculated as:

$$\text{Score}(i, w) = w_L \Delta \widehat{D}_L(i, w) + w_R \Delta \widehat{D}_R(i, w) + w_C \Delta \widehat{D}_C(i, w) + w_{hist} \Delta \widehat{D}_{hist}(i, w) + w_{rot} \Delta \widehat{D}_{rot}(i, w) \quad (51)$$

A negative marginal value indicates that the candidate assignment improves the corresponding objective relative to the current state. A positive value indicates that the assignment increases the respective deviation.

The volume component measures the effect of the assignment on the normalized warehouse-volume deviation. The pallet ratio component evaluates the change in proportional pallet allocation. The category component measures whether the assignment moves current category shares toward or away from their target values.

The historical component evaluates the change in the signed projected imbalance. An assignment to an historically over-allocated category–location combination generally increases this component. Conversely, an assignment to an historically under-allocated combination may reduce the absolute imbalance and therefore decrease the score.

The rotation component evaluates the consistency of the candidate assignment with the preference signal generated by the strategic policy and rotation layer.

All components are dimensionless because they are normalized in Section 4. The objective weights can therefore be interpreted as relative operational priorities rather than numerical corrections for incompatible physical units.

Capacity does not appear as an additional scoring term. Capacity and compatibility are hard feasibility conditions and are enforced before the scoring process through Equation (47).

The objective weights remain fixed during an experimental run. Adaptive behavior results from changes in the operational state and historical imbalance state rather than from repeated modification of the objective weights. This distinguishes the proposed mechanism from conventional adaptive penalty-update methods.

5.6. Weight Selection

The objective weights determine the relative importance of shipment volume balance, pallet ratio consistency, category distribution, historical correction, and strategic rotation adherence. Because the corresponding objective components are normalized, the weights can be interpreted directly as relative operational priorities.

The final objective-weight vector used in the experimental evaluation is:

$$(w_L, w_R, w_C, w_{hist}, w_{rot}) = (0.25, 0.20, 0.20, 0.25, 0.10)$$

The comparatively high weights assigned to shipment volume balance and historical correction reflect the operational objective of combining short-term workload balance with persistent correction of category–warehouse imbalances. The pallet ratio and category distribution components receive equal intermediate weights, while the strategic rotation term acts as a supplementary soft preference and therefore receives a lower weight.

The complexity-ordering weights are fixed as:

$$(\lambda_F, \lambda_G, \lambda_P, \lambda_L) = (0.40, 0.20, 0.25, 0.15)$$

The largest weight is assigned to the number of feasible warehouse alternatives because deliveries with limited assignment flexibility should be processed first. Pallet quantity receives the second-largest weight because it directly consumes the principal hard capacity resource. Category restrictions and shipment volume provide additional indicators of operational complexity.

Both weight vectors were specified before the final evaluation and remained unchanged across all scenarios, benchmark procedures, and repetitions. The sensitivity analysis in Section 6.7 examines whether moderate changes in the historical feedback weight and the historical-state parameters materially affect the reported allocation behavior.

5.7. Assignment Rule and Tie-Breaking

For each delivery i , the minimum candidate score is defined as:

$$s_i^{\min} = \min_{w \in F_i(S_k)} (i, w)_{\text{Score}} \quad (52)$$

To avoid unstable decisions caused by negligible numerical differences, a near-tie candidate set is introduced:

$$Ti = \left\{ w \in F_i(S_k) \mid \text{Score}(i, w) \leq s_i^{\min} + \varepsilon_{\text{tie}} \right\} \quad (53)$$

If the near-tie set contains a single warehouse, that warehouse is selected directly. If several warehouses belong to the set, the algorithm applies a deterministic lexicographic tie-breaking rule.

The projected maximum resource utilization after assigning delivery i to warehouse w is calculated as:

$$u^{\text{proj}}_{i,w} = \max r \in R \left(\frac{C_{w,r,t} - C^{\text{rem}}_{w,r} + q_{i,r}}{\max(C_{w,r,t}, \varepsilon)} \right) \quad (54)$$

The delivery-specific strategic rotation preference is defined as:

$$\pi_{i,w,t} = \frac{\sum_{c \in C} b_{i,c} \pi_{c,w,t}}{\max(\sum_{c \in C} b_{i,c}, \varepsilon)} \quad (55)$$

Candidate warehouses are compared according to the following sequence:

1. Lowest projected maximum resource utilization;

2. Highest delivery-specific rotation preference; and
3. Lowest warehouse index.

The resulting selection rule is expressed as:

$$w^* = \text{lexmin} w \in \mathcal{T}i(u^{\text{Proj}}i, w, -\pi i, w, t, w) \quad (56)$$

The assignment variables are subsequently updated through:

$$x_{i,w^*} = 1, x_{i,w} = 0 \quad \forall w \in J \setminus \{w^*\}, y_i = 0 \quad (57)$$

The deterministic tie-breaking procedure ensures reproducibility across repeated runs with identical input data and parameter values.

5.8. Sequential State Update

Following the assignment of delivery i to warehouse w^* , all affected state variables are updated before the next delivery is evaluated.

The remaining resource capacities are updated through:

$$C^{\text{rem}}_{w^*,r} \leftarrow C^{\text{rem}}_{w^*,r} - q_{i,r} \quad \forall r \in R \quad (58)$$

The cumulative shipment volume is updated as:

$$V_{w^*} \leftarrow V_{w^*} + L_i \quad (59)$$

The cumulative pallet quantity is updated as:

$$P_{w^*} \leftarrow P_{w^*} + P_i \quad (60)$$

The category-balancing quantities are updated through:

$$B_{c,w^*} \leftarrow B_{c,w^*} + b_{i,c} \quad \forall c \in C \quad (61)$$

The updated operational state is used immediately when evaluating the next delivery. Consequently, each assignment reflects all preceding decisions made within the same daily batch-oriented planning run.

The historical state derived from previously completed daily periods provides the initial historical condition for the current run. After each individual assignment, the current-day category quantities and the projected historical imbalance are updated and used when scoring the next delivery. Thus, the internal allocation state evolves sequentially even though the complete algorithm is executed only once per day for the full set of available deliveries.

After completion of the daily allocation process, the realized category shares and signed deviations are calculated according to Equations (22) and (23). These final daily deviations are added once to the rolling historical record. If the updated record contains more than 30 completed daily periods, the oldest observation is removed. The resulting rolling window is then used to calculate the historical state for the subsequent planning day according to Equation (26).

If a delivery enters the exception procedure, the remaining warehouse capacities and cumulative allocation quantities remain unchanged. Only the corresponding exception indicator and operational exception record are updated.

5.9. Complete Algorithm

Figure 2 summarizes the conceptual decision workflow of the proposed framework, whereas Algorithm 1 specifies the complete computational procedure used for implementation and replication. The algorithm explicitly defines initialization, complexity-based delivery ordering, dynamic feasibility filtering, normalized marginal scoring, deterministic tie-breaking, exception handling, sequential state updating, and end-of-cycle historical updating.

Algorithm 1. Sequential normalized marginal-scoring allocation procedure

Input: delivery set I_t ; warehouse set J ; resource set R ; category set C ; shipment attributes $L_i, P_i, q_{i,r}, g_{i,c}$, and $b_{i,c}$; capacities $C_{w,r,t}$; compatibility indicators $a_{i,w}$; target shares $\alpha_{w,t}, \beta_{w,t}$, and $\gamma_{c,w}$; rotation preferences $\pi_{c,w,t}$; historical states $h_{c,w,t}$; objective weights w_L, w_R, w_C, w_{hist} , and w_{rot} ; ordering weights $\lambda_F, \lambda_G, \lambda_P$, and λ_L ; rolling-window length $H = 30$ daily planning periods; and tie tolerance ε_{tie} .

Output: assignment variables $x_{i,w}$; exception indicators y_i ; final warehouse states; realized allocation metrics; updated historical state $h_{c,w,t+1}$; exception records; and computational runtime.

- 1: Start the runtime timer.
 - 2: Initialize remaining capacities $C_{w,r}^{rem} \leftarrow C_{w,r,t}$ for all $w \in J$ and $r \in R$.
 - 3: Initialize $V_w \leftarrow 0$, $P_w \leftarrow 0$, and $B_{c,w} \leftarrow 0$ for all applicable warehouses and categories.
 - 4: Initialize $x_{i,w} \leftarrow 0$, $y_i \leftarrow 0$, and an empty exception queue.
 - 5: for each delivery $i \in I_t$ do
 - 6: Determine the initial feasible warehouse set F_i^0 according to Equation (43).
 - 7: Calculate the operational-complexity index κ_i according to Equation (44).
 - 8: end for
 - 9: Sort deliveries in descending order of κ_i and apply the deterministic secondary ordering.
 - 10: for each delivery i in the ordered sequence do
 - 11: Determine the dynamic feasible set $F_i(S_k)$ according to Equation (47).
 - 12: if $F_i(S_k) = \emptyset$ then
 - 13: Set $y_i \leftarrow 1$ and $x_{i,w} \leftarrow 0$ for all $w \in J$.
 - 14: Append delivery i to the operational exception queue.
 - 15: Continue with the next delivery.
 - 16: end if
 - 17: for each warehouse $w \in F_i(S_k)$ do
 - 18: Construct the projected state $S_k \oplus (i,w)$.
 - 19: Calculate the normalized marginal objective changes according to Equation (50).
 - 20: Calculate Score (i,w) according to Equation (51).
 - 21: end for
 - 22: Determine the minimum candidate score s_i^{min} according to Equation (52).
 - 23: Construct the near-tie set T_i according to Equation (53).
 - 24: if $|T_i| = 1$ then
 - 25: Select the unique warehouse $w^* \in T_i$.
 - 26: else
 - 27: Calculate projected resource utilization and delivery-specific rotation preference.
 - 28: Select w^* using the lexicographic rule in Equation (56).
 - 29: end if
 - 30: Apply the assignment decision according to Equation (57).
 - 31: Update the remaining capacities, cumulative allocation quantities, current-day category quantities, and projected historical imbalance before processing the next delivery according to Equations (58)–(61).
 - 32: end for
 - 33: Calculate the realized warehouse, category, and exception metrics.
 - 34: Calculate the signed category–warehouse deviations for the completed planning cycle.
 - 35: After all deliveries have been processed, add the final daily deviations to the rolling historical record and remove the oldest observation if the record exceeds 30 completed daily periods.
 - 36: Calculate the historical state for cycle $t + 1$ according to Equation (26).
 - 37: Stop the runtime timer and record the computational time.
 - 38: Return assignments, exception records, final states, allocation metrics, updated historical observations, and runtime.
-

Provided that the input data, parameter values, ordering rules, and tie tolerance remain unchanged, Algorithm 1 produces deterministic and reproducible allocation results. Hard capacity and compatibility constraints are never relaxed within the automatic procedure. Deliveries without a feasible warehouse are transferred to the documented exception process rather than being assigned through a compensating soft penalty.

In the evaluated implementation, Algorithm 1 is executed once per day and the rolling historical record is limited to the most recent 30 completed daily allocation periods.

5.10. Computational Complexity

The initial feasibility analysis requires:

$$O(|I_t||J||R|)$$

operations.

Calculation of all complexity-index values requires:

$$O(|I_t|(|J||R| + |C|))$$

operations.

Sorting the delivery set requires:

$$O(|I_t|\log|I_t|)$$

operations.

During sequential allocation, each delivery is evaluated for at most $|J|$ warehouse locations. Using incremental warehouse and category states, evaluation of one feasible candidate requires approximately:

$$O(|R| + |C_i|)$$

operations, where $|C_i|$ denotes the number of active categories associated with delivery i .

The resulting overall time complexity is:

$$O(|I_t|\log|I_t| + |I_t||J|(|R| + \overline{|C_i|})) \quad (62)$$

where the term $\overline{|C_i|}$ denotes the average number of active categories per delivery.

For a fixed and comparatively small number of warehouses, resource dimensions, and shipment categories, the computational effort therefore increases approximately linearly with the number of deliveries.

The memory requirement is:

$$O(|I_t| + |J||R| + |J||C|) \quad (63)$$

The computational behavior was evaluated using daily allocation instances ranging from 100 to 1500 pallets. Across all evaluated instance sizes, the complete procedure was completed in less than 5 s. Because the allocation process is executed once per day for the subsequent operational period, this runtime is substantially shorter than the available planning window.

5.11. Methodological Interpretation

The proposed procedure is a multi-objective constructive heuristic rather than a conventional single-criterion greedy algorithm. Each local assignment decision is derived

from normalized marginal changes in the global objective function and therefore reflects current warehouse utilization, category distribution, strategic preferences, and historical imbalance simultaneously.

The sequential procedure does not guarantee global optimality because assignments are generated successively without unrestricted backtracking. Its solution quality is therefore evaluated in Section 6 against non-adaptive, residual capacity, static-priority, and random-feasible benchmark procedures. In addition, five reduced 100-pallet instances are solved using the exact MILP formulation to quantify the relative objective gap of the heuristic. Larger-scale exact comparisons remain a subject for future research.

The method nevertheless preserves all hard feasibility constraints, generates deterministic recommendations, and provides predictable computational behavior. Every assignment can be decomposed into its individual objective contributions, enabling planners to identify whether a recommendation is primarily influenced by capacity utilization, pallet distribution, category balance, strategic rotation, or historical correction.

This combination of normalized objective-based scoring, signed historical feedback, explicit operational policy, deterministic tie-breaking, and exception handling provides the methodological distinction between the proposed procedure and conventional greedy allocation, adaptive penalty methods, and rolling-horizon reoptimization.

6. Experimental Evaluation and Discussion

6.1. Evaluation Objectives

The experimental evaluation examines the operational feasibility, balancing quality, historical correction behavior, computational scalability, and practical applicability of the proposed framework for daily next-day allocation. The evaluation addresses the following questions:

- EQ1: Does the proposed allocation procedure preserve all hard warehouse capacity and delivery compatibility constraints across the evaluated scenarios?
- EQ2: Does the framework maintain balanced warehouse utilization, pallet distribution, and category allocation as the daily workload increases?
- EQ3: Does the signed historical feedback mechanism reduce persistent category–warehouse imbalances over the rolling window of 30 consecutive daily planning periods?
- EQ4: Does the allocation procedure remain computationally scalable for daily instances ranging from 100 to 1500 pallets, with approximately 130 lots per pallet on average?
- EQ5: Can the complete allocation process be completed within the available planning window for next-day operations?

The evaluation combines warehouse-level utilization analysis, deviation-based balancing metrics, the temporal development of the historical imbalance components, and the observed execution-time bound of less than 5 s.

6.2. Scenario Generation and Experimental Protocol

The experimental evaluation uses a parameterized computational environment representing a six-warehouse inbound allocation system. Warehouse identities are anonymized as W1–W6. The evaluation reproduces the operational structure of the investigated application, including non-divisible deliveries, pallet-based capacity consumption, shipment categories, warehouse-specific compatibility restrictions, and category-dependent allocation preferences.

Two complementary experimental dimensions are considered. First, the balancing behavior of the framework is examined under four relative operational demand conditions: low volume (70% of nominal demand), nominal volume (100%), high volume (130%), and

very high volume (150%). These scenarios are used to evaluate warehouse utilization and allocation balance.

Second, computational scalability is evaluated using daily instances ranging from 100 to 1500 pallets. Each pallet is associated with approximately 130 lots on average, where the term lot follows the operational unit definition used in the underlying data. The allocation algorithm is executed once per day and produces the warehouse plan for the following operational day.

The historical mechanism is evaluated over 30 consecutive daily planning periods. After each daily run, the realized category–warehouse deviations are incorporated into a rolling 30-day historical window. Observations older than 30 completed periods are removed before the state for the next planning day is calculated.

Six anonymized warehouses, W1–W6, provide a combined daily capacity of 1500 pallet positions. Four representative demand conditions are defined relative to a nominal 1000-pallet day: 700 pallets (70%), 1000 pallets (100%), 1300 pallets (130%), and 1500 pallets (150%). Additional scalability tests use 100, 250, 500, 750, 1000, 1250, and 1500 pallets. Delivery sizes follow a rounded triangular distribution with minimum 1, mode 12, and maximum 30 pallets. Lots per pallet follow a truncated normal distribution with mean 130, standard deviation 15, and bounds of 80 and 180. These explicit assumptions preserve the reported average while producing heterogeneous deliveries.

Five shipment categories are generated with probabilities of 0.50 for standard goods, 0.15 for heavy goods, 0.15 for seasonal goods, 0.12 for beverages, and 0.08 for hazardous or special-handling goods. Compatibility is category-driven and makes approximately 17% of delivery–warehouse pairs infeasible. Every generated delivery is initially compatible with at least one warehouse. Category balance is evaluated in pallet units, so that $b_{i,c} = P_{ig_{i,c}}$. The same random seeds are reused across benchmark methods to enable paired comparisons (Table 4).

Table 4. Experimental configuration used for scenario generation, benchmark evaluation, and runtime assessment.

Parameter	Setting	Role in the Experiment
Warehouses	6 locations (W1–W6)	Multi-location allocation network
Daily capacities	W1 = 240, W2 = 260, W3 = 250, W4 = 255, W5 = 245, W6 = 250 pallets; total = 1500	Hard pallet capacity limits
Scalability range	100, 250, 500, 750, 1000, 1250, and 1500 pallets/day	Runtime and scale assessment
Demand scenarios	700, 1000, 1300, and 1500 pallets/day	70%, 100%, 130%, and 150% of nominal demand
Lots per pallet	Truncated normal; mean = 130, SD = 15, bounds = 80–180	Operational workload proxy
Delivery size	Rounded triangular distribution Triangular (1, 12, 30) pallets	Non-divisible inbound deliveries
Shipment categories	5 categories with probabilities 0.50/0.15/0.15/0.12/0.08	Category-balancing test
Compatibility	Category-driven; approximately 17% infeasible pairs	Warehouse-specific handling restrictions
Historical parameters	$H = 30, \rho = 0.85, \eta = 0.30, \bar{h} = 0.25$	Rolling signed historical correction
Fixed expert-informed operational priorities	$(w_L, w_R, w_C, w_{hist}, w_{rot}) = (0.25, 0.20, 0.20, 0.25, 0.10)$	Normalized multi-objective score
Fixed complexity-based delivery ordering priorities	$(\lambda_F, \lambda_G, \lambda_P, \lambda_L) = (0.40, 0.20, 0.25, 0.15)$	Complexity-based delivery sequence

Table 4. Cont.

Parameter	Setting	Role in the Experiment
Scenario repetitions	30 independent runs per representative demand level	Means, standard deviations, and confidence intervals
Sequential assessment	30 consecutive daily planning cycles	Convergence of historical feedback
Exact benchmark	5 independently generated 100-pallet instances	MILP optimality comparison
Implementation	Python 3.11; heuristic implementation using NumPy 2.2.6/SciPy 1.15.3; exact MILP solved with Gurobi 11.0.3; MIPGap = 0.1%; time limit = 600 s; Intel Core i7-12700H; 32 GB RAM; Windows 11 (64-bit).	Reproducible computational implementation
Update frequency	Once daily for next-day allocation	Operational deployment cycle

The four representative demand scenarios are evaluated using 30 independent seeds. Reported values are means $\pm$ standard deviations; two-sided 95% confidence intervals are calculated as $\bar{x} \pm t_{0.975, n-1} s / \sqrt{n}$. The exact MILP benchmark uses five smaller instances because the purpose of the exact model is to quantify the heuristic gap rather than to replace the daily operational procedure. The deterministic scenarios are solved individually; sample-average approximation is therefore not required because uncertainty is represented through repeated out-of-sample scenarios rather than within a stochastic expected-value objective.

6.3. Daily Allocation and Implementation Protocol

The allocation procedure is executed once per day after the inbound information required for the subsequent operational period has become available. Each daily run performs data initialization, complexity-based delivery ordering, feasibility filtering, normalized marginal scoring, deterministic tie-breaking, exception handling, and final state updating.

The generated assignments are stored as the allocation plan for the following operational day. After completion of the run, the realized category–warehouse deviations are added to the historical record. The record contains at most 30 completed daily periods and is subsequently used to calculate the historical correction state for the next run.

Two runtime measures are distinguished. Core algorithm runtime covers allocation-state initialization, complexity-based delivery ordering, feasibility filtering, normalized marginal scoring, assignment, and in-memory state updating. End-to-end operational runtime additionally includes input loading, data preparation, result generation, output persistence, and updating of the rolling historical record. The core algorithm runtimes are reported in Tables 5 and 6, whereas the complete end-to-end operational update remained below 5 s across all evaluated instances.

Table 5. Scenario results over 30 independent replications; entries are mean $\pm$ standard deviation.

Scenario	Input Pallets	Deliveries	Utilization (%)	Spread (pp)	Exceptions (%)	Core Algorithm Runtime (s)
Low (70%)	700	50.20 $\pm$ 2.76	46.66 $\pm$ 0.01	3.63 $\pm$ 1.80	0.000 $\pm$ 0.000	0.038 $\pm$ 0.004
Nominal (100%)	1000	69.40 $\pm$ 3.38	66.66 $\pm$ 0.02	4.39 $\pm$ 1.73	0.000 $\pm$ 0.000	0.052 $\pm$ 0.004
High (130%)	1300	90.87 $\pm$ 4.40	86.66 $\pm$ 0.02	6.24 $\pm$ 2.54	0.000 $\pm$ 0.000	0.071 $\pm$ 0.008
Very high (150%)	1500	106.40 $\pm$ 4.77	99.64 $\pm$ 0.14	1.07 $\pm$ 0.41	0.358 $\pm$ 0.142	0.075 $\pm$ 0.010

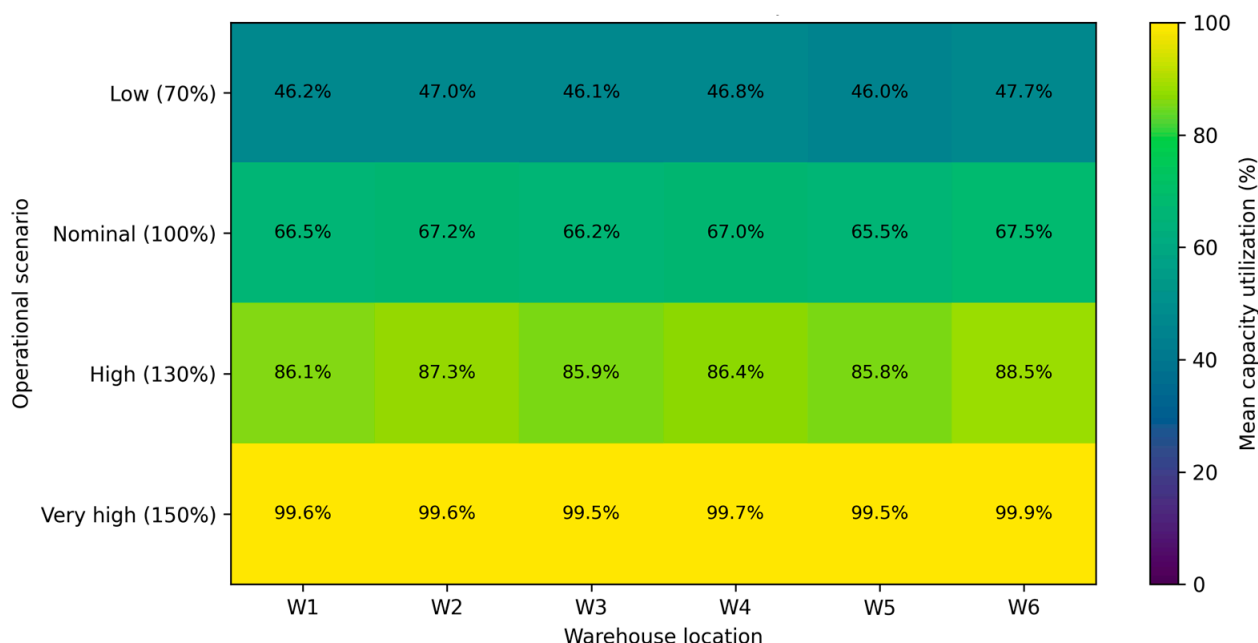
Table 6. Benchmark results for nominal 1000-pallet scenarios over 30 paired replications; entries are mean $\pm$ standard deviation.

Method	Volume Dev. (%)	Pallet Dev. (%)	Category Dev. (%)	Balance Index	Exceptions (%)	Runtime (s)
Proposed adaptive	1.40 $\pm$ 0.59	1.36 $\pm$ 0.55	15.93 $\pm$ 4.36	0.982 $\pm$ 0.008	0.000 $\pm$ 0.000	0.052 $\pm$ 0.003
Non-adaptive ablation	1.16 $\pm$ 0.63	1.40 $\pm$ 0.55	15.92 $\pm$ 4.59	0.983 $\pm$ 0.007	0.000 $\pm$ 0.000	0.051 $\pm$ 0.003
Residual capacity greedy	3.02 $\pm$ 0.97	0.77 $\pm$ 0.38	48.12 $\pm$ 9.80	0.992 $\pm$ 0.004	0.000 $\pm$ 0.000	0.001 $\pm$ 0.000
Static priority	64.67 $\pm$ 0.00	64.67 $\pm$ 0.00	105.57 $\pm$ 4.88	0.500 $\pm$ 0.000	0.000 $\pm$ 0.000	0.001 $\pm$ 0.000
Random feasible	26.85 $\pm$ 8.74	26.19 $\pm$ 8.40	47.48 $\pm$ 9.33	0.711 $\pm$ 0.073	0.000 $\pm$ 0.000	0.001 $\pm$ 0.000

6.4. Scenario-Based Utilization and Feasibility

Table 5 summarizes the four representative demand conditions. Mean network utilization increases almost proportionally with inbound pallet volume. No exceptions occur at 700, 1000, or 1300 pallets. At 1500 pallets, the mean exception rate is 0.36%, reflecting indivisibility and category-specific compatibility at the physical capacity boundary rather than hidden capacity violations. The algorithm therefore preserves hard feasibility even when the total input equals the combined nominal capacity. The exception rate is defined as the percentage of deliveries for which $y_i = 1$.

The 95% confidence half-width for mean utilization is below 0.01 percentage points in the first three scenarios and 0.05 percentage points at maximum load. The average utilization spread remains below 6.3 percentage points in all scenarios and decreases to 1.07 percentage points when the network approaches full utilization. Figure 3 reports the mean warehouse-level utilization values obtained from 30 independent replications for each of the four representative demand scenarios.

**Figure 3.** Mean warehouse-level utilization across the four representative demand scenarios, averaged over 30 independent replications.

6.5. Benchmark and Exact-Model Comparison

Table 6 compares the proposed procedure with the non-adaptive, residual capacity, static-priority, and random-feasible benchmarks for nominal 1000-pallet days. The proposed and non-adaptive procedures produce similarly low immediate volume and pallet deviations, indicating that the historical term does not materially degrade short-term balancing. The residual capacity rule produces a slightly lower pallet deviation but substantially poorer category balance. Static and random rules perform considerably worse across

all balancing indicators. This confirms that the improvement is generated by normalized multi-objective scoring rather than feasibility filtering alone. To quantify the equality of pallet capacity utilization across warehouse locations, a Jain-type balance index is calculated. Let u_w denote the realized pallet utilization of warehouse w :

$$u_w = \frac{\sum_{i \in I_t} P_i x_{i,w}}{C_{w,p,t}}$$

The warehouse balance index is defined as:

$$BI = \frac{(\sum_{w \in J} u_w)^2}{|J| \sum_{w \in J} u_w^2} \quad (64)$$

The balance index lies between $1/|J|$ and 1. A value of 1 indicates identical capacity-utilization levels across all warehouses, whereas lower values indicate increasing concentration of workload at individual locations. The index evaluates only the equality of warehouse utilization and does not incorporate category distribution or historical imbalance deviations. Consequently, a method may achieve a high balance index while still producing an unfavorable category distribution.

For the exact benchmark, the relative objective gap between the sequential heuristic and the MILP solution is calculated as:

$$\text{Gap} = \frac{Z_{\text{heur}} - Z_{\text{MILP}}}{\max(|Z_{\text{MILP}}|, \epsilon)} \times 100\% \quad (65)$$

The same normalized objective components, objective weights, exception penalty, and input data are used for both methods. Because the problem is formulated as a minimization problem, a gap of 0% indicates that the heuristic and the exact MILP obtain the same objective value, whereas a positive value indicates a higher objective value for the heuristic.

For the five exact 100-pallet benchmark instances, all MILP models were solved to proven optimality. The heuristic produced a mean relative objective gap of 1.10% with a standard deviation of 1.09%. Using the Student-t distribution for $n = 5$ independent instances, the two-sided 95% confidence interval has a half-width of approximately 1.35 percentage points. Mean solution time was 0.005 s for the heuristic and 0.059 s for the exact MILP. These results indicate that the constructive procedure remains close to the exact objective on reduced instances while retaining lower computational overhead.

6.6. Sequential Historical Correction

Figure 4 shows a representative 30-day sequence in which the allocation state is updated once per day and used for next-day planning. Historical imbalance decreases from 6.67% to 1.61%, corresponding to a 75.8% reduction. Volume deviation decreases from 1.01% to 0.49% and category deviation from 13.50% to 11.24%. Pallet ratio deviation remains low throughout and changes from 2.13% to 2.07%.

The category-deviation series fluctuates because the category mix changes from day to day, but the historical state follows a clear downward trajectory. This distinction is important: the feedback mechanism is not intended to eliminate normal daily variation. Its purpose is to prevent persistent directional bias from accumulating across completed planning cycles.

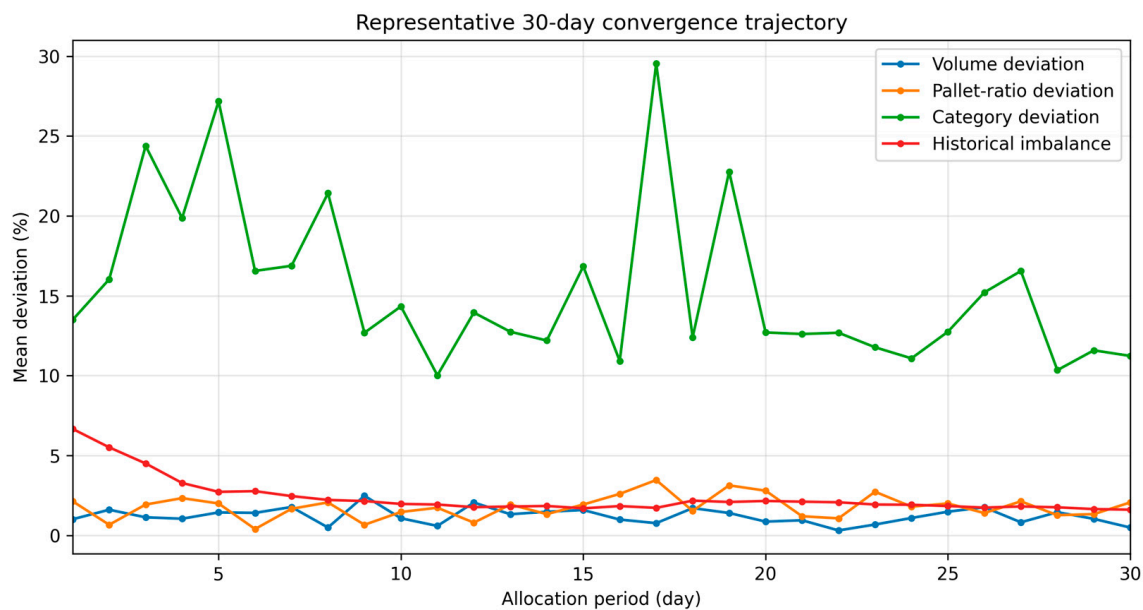


Figure 4. Representative 30-day convergence trajectory for volume, pallet ratio, category, and signed historical imbalance deviations under daily adaptive updating.

6.7. Sensitivity Analysis

To evaluate the temporal consistency of the warehouse allocation, the normalized utilization share of warehouse w in daily planning period t is defined as:

$$s_{w,t} = \frac{u_{w,t}}{\sum_{v \in J} u_{v,t}}$$

Based on the day-to-day changes in these utilization shares, the allocation stability index over T consecutive daily planning periods is defined as:

$$SI = 1 - \frac{1}{2(T-1)} \sum_{t=2}^T \sum_{w \in J} |s_{w,t} - s_{w,t-1}| \quad (66)$$

The stability index ranges from 0 to 1. Values approaching 1 indicate that the relative warehouse allocation structure changes only slightly between consecutive daily planning periods, whereas lower values indicate stronger day-to-day fluctuations. In the sequential evaluation, $T = 30$ daily planning periods.

Sensitivity tests vary the historical window length $H \in \{5, 10, 20, 30\}$, the decay factor $\rho \in \{0.70, 0.85, 0.95\}$, the correction gain $\eta \in \{0.15, 0.30, 0.45\}$, and the historical weight $w_{hist} \in \{0.10, 0.25, 0.40\}$. The operational baseline is defined by $H = 30$, $\rho = 0.85$, $\eta = 0.30$, and $w_{hist} = 0.25$. The baseline configuration $H = 30$, $\rho = 0.85$, and $\eta = 0.30$ yields a volume deviation of 0.86%, category deviation of 22.10%, historical imbalance of 1.96%, and stability index of 0.994 in the controlled sensitivity sequence. Across all tested settings, category deviation remains between 22.08% and 22.60%, historical imbalance between 1.93% and 2.30%, and stability between 0.991 and 0.995.

The selected 30-period window provides the strongest temporal smoothing among the evaluated settings and corresponds to the monthly operational memory used in the investigated application.

6.8. Runtime and Operational Scalability

The allocation algorithm was executed once per day to generate warehouse assignments for the subsequent operational period. The computational experiments covered

daily instances ranging from 100 to 1500 pallets, with approximately 130 lots associated with each pallet on average.

Across all evaluated instance sizes, the complete allocation procedure, including feasibility filtering, normalized marginal scoring, deterministic tie-breaking, exception handling, and state updating, was completed in less than 5 s. This runtime was substantially shorter than the available operational planning window and supports the application of the framework for daily next-day allocation.

After each daily run, the realized category–warehouse deviations were incorporated into a rolling historical window containing the most recent 30 daily planning periods. Observations older than 30 periods were removed before calculating the historical state for the following day. During system initialization, the historical weights were renormalized over the observations available at that time.

6.9. Discussion

The results support three main conclusions. First, feasibility filtering and explicit exception handling prevent hidden capacity violations even at the physical capacity boundary. Second, normalized marginal scoring improves volume and category balance substantially relative to static and random rules, while remaining close to the exact MILP on small instances. Third, the signed historical state provides directional long-term correction without materially compromising immediate pallet and volume balance.

The strongest operational advantage is not a single numerical objective value but the combination of traceability and speed. Each assignment can be decomposed into feasibility, current balance, category distribution, strategic rotation, and historical correction. A planner can therefore understand the reason for a recommendation, while the once-daily update completes well within the available operational window.

6.10. Limitations

The evaluation is based on a parameterized synthetic scenario generator calibrated to the reported operational scale rather than on a fully disclosed long-term industrial dataset. The evaluated range of 100 to 1500 pallets, the average workload of approximately 130 lots per pallet, the once-daily planning frequency, and the observed end-to-end runtime of less than 5 s reflect the investigated application context. However, the detailed delivery-size, lot-density, category, and warehouse-compatibility distributions were generated according to explicit simulation assumptions. This distinction is reported transparently to avoid presenting synthetically generated observations as directly measured industrial data.

The historical evaluation covers 30 consecutive daily planning periods and therefore demonstrates the behavior of the complete rolling historical window used in the operational implementation. Nevertheless, this period does not constitute long-term field validation across seasonal demand changes, structural network modifications, or extended operational disruptions.

The exact optimization benchmark is limited to five instances at the 100-pallet scale. Consequently, the reported comparison does not establish global optimality or a general optimality-gap bound for larger operational instances. Exact comparisons at higher pallet volumes may require substantially longer solution times, larger computational resources, or problem-specific decomposition techniques.

The current framework also excludes transportation routing, dock scheduling, workforce availability, queueing effects, service-time variability, and uncertain delivery arrival times. Furthermore, the runtime result of less than 5 s was observed on the specified test system and should not be interpreted as a hardware-independent upper bound.

Future research may validate the framework using longer industrial observation periods and larger exact benchmark sets. Further extensions may combine the proposed allocation logic with discrete-event simulation, stochastic programming, sample-average approximation, predictive arrival models, transportation and dock synchronization, or learning-based parameter adaptation while preserving hard feasibility, interpretability, and operational controllability.

7. Conclusions

This study developed a hierarchical two-level framework for assigning non-divisible inbound deliveries to multiple warehouse locations under heterogeneous operational constraints. The strategic layer generates transparent category–location preferences, while the operational layer applies hard feasibility filtering, normalized marginal scoring, deterministic tie-breaking, and explicit exception handling. A signed rolling historical state distinguishes over-allocation from under-allocation and transfers corrective information across daily planning cycles.

The computational evaluation covered 100–1500 pallets per day, approximately 130 lots per pallet, six warehouses, and once-daily updates for next-day planning. Across 30 replications of the representative demand scenarios, mean network utilization increased from 46.66% at 700 pallets to 99.64% at 1500 pallets. No exceptions occurred below the physical capacity boundary; at 1500 pallets, exceptions averaged only 0.36% and were handled without violating warehouse capacity or compatibility constraints.

The proposed multi-objective scoring procedure substantially outperformed static and random allocation rules and produced markedly better category balance than a residual capacity greedy method. On five 100-pallet instances, the heuristic remained within 1.10% of the exact MILP objective on average. In the representative 30-day sequence, signed historical imbalance decreased by 75.8%, while volume and category deviations also declined. These results demonstrate that the feedback mechanism supports long-term correction without sacrificing short-term feasibility.

The computational core required between 0.004 and 0.067 s across the tested scale, with a maximum recorded value of 0.080 s. Including daily data preparation, state updating, output generation, and persistence, the complete operational update remained below 5 s. Since the procedure is executed once per day, the available planning window is therefore substantially larger than the required runtime.

The main limitation is the use of a calibrated synthetic scenario generator rather than a fully disclosed long-term industrial dataset. Further work should validate the method on extended operational records, increase the size of exact benchmark instances, and incorporate routing, dock scheduling, labor, congestion, and arrival time uncertainty. Stochastic optimization and discrete-event simulation may complement the current deterministic daily allocation model.

Overall, the framework provides an interpretable and scalable link between global allocation objectives and practical daily warehouse decisions. Its combination of hierarchical policy guidance, normalized shipment-level scoring, explicit exception handling, and signed historical correction offers a reproducible foundation for adaptive multi-location inbound logistics planning.

Author Contributions: Conceptualization, M.H.; methodology, M.H.; software, M.H.; validation, M.H.; investigation, M.H.; resources, M.H.; writing—original draft preparation, M.H.; writing—review and editing, M.H. and B.N.; supervision, B.N. All authors have read and agreed to the published version of the manuscript.

Funding: The APC was funded by Universität Duisburg–Essen. We acknowledge support from the Open Access Publication Fund of the University of Duisburg–Essen.

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. The parameterized synthetic scenario-generation rules, fixed parameter values, and evaluation protocol are reported in Section 6. Further in-quiries can be directed to the corresponding authors.

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

References

1. Simchi-Levi, D.; Kaminsky, P.; Simchi-Levi, E. *Designing and Managing the Supply Chain: Concepts, Strategies, and Case Studies*, 3rd ed.; McGraw-Hill: New York, NY, USA, 2008.
2. Snyder, L.V.; Shen, Z.-J.M. *Fundamentals of Supply Chain Theory*, 2nd ed.; Wiley: Hoboken, NJ, USA, 2019. [\[CrossRef\]](#)
3. Silver, E.A.; Pyke, D.F.; Peterson, R. *Inventory Management and Production Planning and Scheduling*, 3rd ed.; Wiley: New York, NY, USA, 1998.
4. Tempelmeier, H. *Inventory Management in Supply Networks: Problems, Models, Solutions*, 2nd ed.; Books on Demand: Norderstedt, Germany, 2011.
5. Melo, M.T.; Nickel, S.; Saldanha-da-Gama, F. Facility location and supply chain management—A review. *Eur. J. Oper. Res.* **2009**, *196*, 401–412. [\[CrossRef\]](#)
6. Shapiro, J.F. *Modeling the Supply Chain*, 2nd ed.; Thomson Brooks/Cole: Belmont, CA, USA, 2007.
7. Stadler, H. Supply chain management and advanced planning—Basics, overview and challenges. *Eur. J. Oper. Res.* **2005**, *163*, 575–588. [\[CrossRef\]](#)
8. Chopra, S.; Meindl, P. *Supply Chain Management: Strategy, Planning, and Operation*, 7th ed.; Pearson: Boston, MA, USA, 2019.
9. Bertsimas, D.; Weismantel, R. *Optimization Over Integers*; Dynamic Ideas: Belmont, MA, USA, 2005.
10. Dantzig, G.B. *Linear Programming and Extensions*; Princeton University Press: Princeton, NJ, USA, 1963.
11. Birge, J.R.; Louveaux, F. *Introduction to Stochastic Programming*, 2nd ed.; Springer: New York, NY, USA, 2011. [\[CrossRef\]](#)
12. Nocedal, J.; Wright, S.J. *Numerical Optimization*, 2nd ed.; Springer: New York, NY, USA, 2006. [\[CrossRef\]](#)
13. Ahuja, R.K.; Magnanti, T.L.; Orlin, J.B. *Network Flows: Theory, Algorithms, and Applications*; Prentice Hall: Upper Saddle River, NJ, USA, 1993.
14. Fatthi, W.N.A.W.A.; Shuib, A.; Dom, R.M. A mixed integer programming model for solving real-time truck-to-door assignment and scheduling problem at cross docking warehouse. *J. Ind. Manag. Optim.* **2016**, *12*, 431–447. [\[CrossRef\]](#)
15. Vidal, T.; Crainic, T.G.; Gendreau, M.; Prins, C. Heuristics for multi-attribute vehicle routing problems: A survey and synthesis. *Eur. J. Oper. Res.* **2013**, *231*, 1–21. [\[CrossRef\]](#)
16. Pisinger, D.; Ropke, S. Large neighborhood search. In *Handbook of Metaheuristics*; Gendreau, M., Potvin, J.-Y., Eds.; Springer: Boston, MA, USA, 2010; pp. 399–419. [\[CrossRef\]](#)
17. Ropke, S.; Pisinger, D. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transp. Sci.* **2006**, *40*, 455–472. [\[CrossRef\]](#)
18. Vidal, T.; Crainic, T.G.; Gendreau, M.; Prins, C. A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time windows. *Comput. Oper. Res.* **2013**, *40*, 475–489. [\[CrossRef\]](#)
19. Xin, J.; Yuan, Q.; D’Ariano, A.; Guo, G.; Liu, Y.; Zhou, Y. Dynamic unbalanced task allocation of warehouse AGVs using integrated adaptive large neighborhood search and Kuhn–Munkres algorithm. *Comput. Ind. Eng.* **2024**, *195*, 110410. [\[CrossRef\]](#)
20. Speranza, M.G. Trends in transportation and logistics. *Eur. J. Oper. Res.* **2018**, *264*, 830–836. [\[CrossRef\]](#)
21. Büyüközkan, G.; Göçer, F. Digital supply chain: Literature review and a proposed framework for future research. *Comput. Ind.* **2018**, *97*, 157–177. [\[CrossRef\]](#)
22. Wang, G.; Gunasekaran, A.; Ngai, E.W.T.; Papadopoulos, T. Big data analytics in logistics and supply chain management: Certain investigations for research and applications. *Int. J. Prod. Econ.* **2016**, *176*, 98–110. [\[CrossRef\]](#)
23. Ivanov, D.; Dolgui, A. A digital supply chain twin for managing the disruption risks and resilience in the era of Industry 4.0. *Prod. Plan. Control* **2021**, *32*, 775–788. [\[CrossRef\]](#)
24. Seyedan, M.; Mafakheri, F. Predictive big data analytics for supply chain demand forecasting: Methods, applications, and research opportunities. *J. Big Data* **2020**, *7*, 53. [\[CrossRef\]](#)
25. Baryannis, G.; Validi, S.; Dani, S.; Antoniou, G. Supply chain risk management and artificial intelligence: State of the art and future research directions. *Int. J. Prod. Res.* **2019**, *57*, 2179–2202. [\[CrossRef\]](#)
26. Tao, F.; Qi, Q.; Liu, A.; Kusiak, A. Data-driven smart manufacturing. *J. Manuf. Syst.* **2018**, *48*, 157–169. [\[CrossRef\]](#)

27. Rai, R.; Tiwari, M.K.; Ivanov, D.; Dolgui, A. Machine learning in manufacturing and Industry 4.0 applications. *Int. J. Prod. Res.* **2021**, *59*, 4773–4778. [[CrossRef](#)]
28. Boute, R.N.; Gijsbrechts, J.; van Jaarsveld, W.; Vanvuchelen, N. Deep reinforcement learning for inventory control: A roadmap. *Eur. J. Oper. Res.* **2022**, *298*, 401–412. [[CrossRef](#)]
29. Gijsbrechts, J.; Boute, R.N.; Van Mieghem, J.A.; Zhang, D. Can deep reinforcement learning improve inventory management? Performance on lost sales, dual-sourcing, and multi-echelon problems. *Manuf. Serv. Oper. Manag.* **2022**, *24*, 1349–1368. [[CrossRef](#)]
30. Oroojlooyjadid, A.; Nazari, M.; Snyder, L.V.; Takáč, M. A deep Q-network for the beer game: Deep reinforcement learning for inventory optimization. *Manuf. Serv. Oper. Manag.* **2022**, *24*, 285–304. [[CrossRef](#)]
31. Dolgui, A.; Ivanov, D. Ripple effect and supply chain disruption management: New trends and research directions. *Int. J. Prod. Res.* **2021**, *59*, 102–109. [[CrossRef](#)]
32. Culot, G.; Nassimbeni, G.; Podrecca, G.; Sartor, M. Artificial intelligence in supply chain management: A systematic literature review of empirical studies and research directions. *Comput. Ind.* **2024**, *162*, 104132. [[CrossRef](#)]
33. Su, Z.; Li, J.; Pang, Q.; Su, M. China futures market and world container shipping economy: An exploratory analysis based on deep learning. *Res. Int. Bus. Financ.* **2025**, *76*, 102870. [[CrossRef](#)]
34. Albert, P.-W.; Rönqvist, M.; Lehoux, N. Trends and new practical applications for warehouse allocation and layout design: A literature review. *SN Appl. Sci.* **2023**, *5*, 378. [[CrossRef](#)]
35. Shi, Q.; Jiang, Y.; Chu, J. Rolling horizon optimization of allocation-location in agricultural emergency supply chains. *Mathematics* **2025**, *13*, 2967. [[CrossRef](#)]
36. Ganbold, O.; Kundu, K.; Li, H.; Zhang, W. A simulation-based optimization method for warehouse worker assignment. *Algorithms* **2020**, *13*, 326. [[CrossRef](#)]
37. Becerra-Fernandez, M.; Romero, O.R.; Trujillo-Diaz, J.; Herrera, M.M. Assignment-simulation model for forklifts in a distribution center with aisle constraints. *Simul. Model. Pract. Theory* **2024**, *133*, 102902. [[CrossRef](#)]

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.