IJMSERH

International Journal of Multidisciplinary and Scientific Emerging ResearcH (IJMSERH)

Volume 12, Issue 3, July-September 2024

Impact Factor: 9.274

ROAD INNO SPACE SJIF Scientific Journal Impact Factor doi Crossref

ijmserh@gmail.com www.ijmserh.com

International Journal of Multidisciplinary and Scientific Emerging ResearcH (IJMSERH)

| ISSN: 2349-6037 | www.ijmserh.com ||Impact Factor: 9.274||Quarterly, Peer Reviewed & Refereed Journal|

|| Volume 12, Issue 3, July-September 2024 ||

DOI: 10.15662/IJMSERH.2024.1203049

Cross-Network Scheduling Optimization for Unified Retail Distribution Networks

Naga Bhuvaneswari Chinnam

Senior Software Engineer, Sam's Club, USA

ABSTRACT: Modern retail distribution networks operate across geographically dispersed distribution centers, fulfillment facilities, transportation hubs, stores, suppliers, and third-party logistics providers. Traditional scheduling approaches frequently optimize these network components independently, resulting in fragmented transportation plans, inefficient resource utilization, unnecessary transfers, and delayed responses to rapidly changing demand and operational conditions. Cross-network scheduling optimization addresses this challenge by coordinating scheduling decisions across multiple interconnected distribution and transportation domains rather than treating each facility as an isolated planning unit. This article presents a generalized architecture for unified retail distribution networks in which demand signals, inventory positions, transportation capacity, warehouse constraints, delivery commitments, and operational events are continuously integrated into a common scheduling framework. The proposed approach combines constraint-based optimization, predictive analytics, event-driven orchestration, dynamic prioritization, and feedback-driven decision mechanisms to improve coordination across the network. A multi-layer architecture is introduced covering data acquisition, network state representation, optimization, execution, monitoring, and continuous learning. The article further examines scheduling objectives such as transportation cost, fulfillment latency, inventory movement, resource utilization, service-level adherence, and network resilience. A generalized optimization formulation demonstrates how competing operational objectives and constraints can be incorporated into cross-network scheduling decisions. The proposed framework also supports incremental adoption by allowing existing warehouse management, transportation management, enterprise resource planning, and order management systems to remain operational while intelligent scheduling capabilities are introduced progressively. By creating a unified decision layer across otherwise fragmented distribution operations, cross-network scheduling can provide a scalable foundation for responsive, resilient, and data-driven retail distribution networks.

KEYWORDS: Cross-network scheduling, retail distribution, supply chain optimization, distribution networks, transportation optimization, warehouse scheduling, fulfillment orchestration, constraint optimization, predictive analytics, event-driven architecture, network resilience, intelligent logistics, inventory optimization, service-level optimization.

I. INTRODUCTION

Retail distribution networks have evolved from relatively linear supply chains into highly interconnected ecosystems involving distribution centers, fulfillment centers, stores, suppliers, transportation providers, cross-docking facilities, regional hubs, and digital commerce channels. The growth of omnichannel retail has further increased the complexity of these networks because customer orders may be fulfilled through multiple pathways, including centralized distribution, store fulfillment, direct shipment, and combinations of these models. As a result, scheduling decisions made at one location can directly influence capacity, inventory availability, transportation requirements, and service performance at other locations.

The increasing availability of real-time operational data provides an important foundation for such coordination. Modern retail environments continuously generate information from order management platforms, warehouse systems, transportation systems, enterprise resource planning platforms, inventory applications, carrier interfaces, Internet of Things devices, and external demand signals. When these data sources are integrated, a more comprehensive representation of network state can be established. Scheduling decisions can then be adjusted as conditions change rather than remaining fixed according to a plan generated at the beginning of an operating period.

A unified scheduling architecture can also incorporate predictive and optimization technologies. Predictive models can estimate demand patterns, transportation delays, workload fluctuations, and potential capacity constraints, while optimization techniques can evaluate alternative schedules against multiple operational objectives. Depending on the characteristics of the network, these techniques may include mathematical programming, constraint programming,

| ISSN: 2349-6037 | www.ijmserh.com ||Impact Factor: 9.274||Quarterly, Peer Reviewed & Refereed Journal|

|| Volume 12, Issue 3, July-September 2024 ||

DOI: 10.15662/IJMSERH.2024.1203049

heuristic optimization, simulation-based approaches, or hybrid optimization models. The objective is not necessarily to replace existing planning systems but to introduce a coordination layer capable of evaluating cross-network consequences before operational decisions are executed.

Another important characteristic of cross-network scheduling is its ability to respond to events. Retail distribution networks are exposed to continuously changing conditions such as order surges, inventory shortages, carrier capacity changes, facility congestion, transportation disruptions, weather events, and unexpected processing delays. A static schedule can become inefficient when the underlying conditions change. An event-driven scheduling architecture can detect significant changes, reassess affected constraints, and generate revised scheduling recommendations or automated actions while preserving unaffected portions of the existing plan.

Cross-network scheduling is particularly relevant to omnichannel retail because inventory and fulfillment resources may be shared across multiple demand channels. A distribution center supporting store replenishment may also support e-commerce fulfillment, while stores may act as temporary fulfillment nodes for online orders. Under these conditions, scheduling decisions cannot be evaluated solely according to warehouse workload or transportation cost. They must also consider customer commitments, inventory availability, labor capacity, transportation windows, and the opportunity cost of allocating shared resources to competing fulfillment requirements.

A generalized cross-network scheduling framework therefore requires several complementary capabilities. First, it needs a common representation of operational state so that information from different systems can be interpreted consistently. Second, it requires a constraint and optimization layer capable of evaluating alternative scheduling decisions. Third, it needs an orchestration mechanism through which scheduling decisions can be communicated to operational systems. Finally, monitoring and feedback capabilities are required to measure execution outcomes and continuously improve future scheduling decisions.

The architecture should also address the practical reality of enterprise environments in which legacy applications cannot be replaced immediately. Warehouse management systems, transportation management systems, enterprise resource planning platforms, and order management applications may have different data models, interfaces, processing cycles, and technology stacks. A cross-network scheduling capability should therefore be designed as an integration and decision layer that can coexist with existing platforms. API-based integration, event streaming, message queues, data pipelines, and standardized data contracts can provide mechanisms for connecting heterogeneous systems without requiring a complete replacement of the underlying operational infrastructure.

From an optimization perspective, the scheduling problem can be viewed as a multi-objective decision problem. A practical schedule may attempt to minimize transportation and handling costs while simultaneously maximizing service-level compliance, resource utilization, inventory availability, and network resilience. These objectives can conflict with one another. Reducing transportation cost, for example, may require consolidation that increases delivery time, whereas maximizing delivery speed may require additional transportation capacity. Cross-network optimization provides a mechanism for representing these trade-offs explicitly through objective functions, constraints, priorities, and business rules.

The effectiveness of such a framework also depends on governance and explainability. Automated scheduling decisions can influence inventory allocation, transportation assignments, warehouse workloads, and customer fulfillment. Consequently, organizations require visibility into why a particular schedule was selected, which constraints influenced the decision, and what alternatives were considered. Human-in-the-loop mechanisms can be incorporated for high-impact decisions, exceptional conditions, or situations where automated optimization confidence falls below a predefined threshold. This creates a balanced architecture in which automation handles high-volume operational decisions while human planners retain control over strategic exceptions and business-critical scenarios.

This article presents a generalized framework for Cross-Network Scheduling Optimization for Unified Retail Distribution Networks. The discussion examines the evolution from localized scheduling toward network-wide coordination, the architectural components required for unified scheduling, data and event integration mechanisms, optimization models, operational constraints, intelligent scheduling techniques, execution orchestration, monitoring, resilience, and performance measurement. The objective is to establish a technology-neutral model that can be adapted to different retail distribution environments while preserving compatibility with existing enterprise systems.

International Journal of Multidisciplinary and Scientific Emerging ResearcH (IJMSERH)

| ISSN: 2349-6037 | www.ijmserh.com ||Impact Factor: 9.274||Quarterly, Peer Reviewed & Refereed Journal|

|| Volume 12, Issue 3, July-September 2024 ||

DOI: 10.15662/IJMSERH.2024.1203049

II. EVOLUTION AND CHARACTERISTICS OF CROSS-NETWORK RETAIL SCHEDULING

Retail distribution scheduling has historically been implemented as a collection of function-specific planning processes. Warehouse management systems typically optimize receiving, picking, packing, and dispatch activities, whereas transportation management systems focus on carrier selection, shipment consolidation, routing, and delivery execution. Inventory planning and order management systems operate according to their respective planning horizons and business constraints. Although this functional decomposition provides operational specialization, it can also produce locally optimized decisions that are not necessarily optimal from a network-wide perspective.

The increasing complexity of retail distribution networks has created a need to coordinate scheduling decisions across organizational, geographic, and technological boundaries. Cross-network scheduling extends the scheduling domain from individual facilities or functions to an interconnected network of distribution centers, fulfillment facilities, stores, transportation hubs, suppliers, carriers, and customer-facing channels. The objective is to evaluate scheduling decisions according to their downstream and upstream effects rather than considering each activity in isolation.

A. Transition from Facility-Level to Network-Level Scheduling

Traditional facility-level scheduling primarily considers local resource availability, inventory position, workforce capacity, and processing requirements. Transportation planning is subsequently performed using the shipment requirements generated by individual facilities. This sequential approach can create coordination gaps because downstream transportation constraints may not be considered when warehouse schedules are established.

Cross-network scheduling introduces a unified decision perspective in which facility operations, transportation capacity, inventory availability, delivery commitments, and resource constraints are evaluated jointly. A scheduling decision at one node can therefore be assessed according to its impact on connected nodes and transportation links.

The resulting decision model can be expressed conceptually as:
Local Operational State + Network Constraints + Demand Requirements → Network-Level Schedule
This approach changes the scheduling objective from maximizing the performance of an individual operational unit to coordinating multiple operational units within a common network.

B. Network Connectivity and Alternative Fulfillment Paths

Contemporary retail distribution networks contain multiple interconnected nodes and transportation relationships. A generalized network may include suppliers, regional distribution centers, fulfillment centers, cross-docking facilities, stores, and end customers. Products can consequently move through multiple possible fulfillment paths.

For instance, an order may follow a conventional distribution path from a regional distribution center to a store, while an e-commerce order may be fulfilled directly from a fulfillment center. In other scenarios, a store may act as a fulfillment node when inventory and customer proximity make this alternative operationally appropriate.

The availability of multiple fulfillment paths increases scheduling flexibility but also expands the decision space. The scheduling system must potentially determine the appropriate combination of source location, processing sequence, transportation mode, departure time, destination, and resource allocation while satisfying operational constraints.

C. Omnichannel Fulfillment and Shared Resources

The growth of omnichannel retail has increased competition for shared distribution resources. The same inventory pool, warehouse labor, transportation capacity, dock infrastructure, and processing equipment may support multiple demand channels simultaneously.

Consequently, scheduling decisions must account for competing operational requirements. A schedule that maximizes warehouse throughput may create transportation bottlenecks, while a transportation-efficient schedule may increase order processing latency. Similarly, allocating inventory to one channel can affect availability for another channel.

A unified scheduling model can address these interactions by incorporating channel priorities, customer commitments, inventory availability, processing capacity, and transportation constraints within a common optimization framework.

| ISSN: 2349-6037 | www.ijmserh.com ||Impact Factor: 9.274||Quarterly, Peer Reviewed & Refereed Journal|

|| Volume 12, Issue 3, July-September 2024 ||

DOI: 10.15662/IJMSERH.2024.1203049

D. Dynamic Scheduling and Continuous Reoptimization

Retail distribution environments are inherently dynamic. Order volumes, inventory positions, transportation availability, facility workloads, and delivery requirements can change after an initial schedule has been generated. Consequently, static scheduling models can become suboptimal when operating conditions deviate from their original assumptions.

Cross-network scheduling incorporates dynamic rescheduling capabilities that allow the scheduling plan to be revised in response to significant operational events. Relevant events may include transportation delays, inventory shortages, facility capacity changes, equipment failures, carrier constraints, unexpected order surges, or changes in customer delivery commitments.

Rather than recalculating the entire network after every event, a scheduling engine can identify the affected activities and perform targeted reoptimization. This reduces unnecessary schedule disruption while maintaining responsiveness to material changes in network conditions.

E. Multi-Objective Scheduling

Cross-network scheduling represents a multi-objective optimization problem because distribution networks typically operate under competing operational objectives. A generalized optimization model may seek to minimize transportation and handling costs while simultaneously improving service-level adherence, resource utilization, fulfillment latency, and network resilience.

Representative objectives include:

- Transportation and handling cost minimization.
- Fulfillment and transportation lead-time reduction.
- Service-level compliance.
- Warehouse and labor utilization.
- Vehicle and carrier utilization.
- Inventory-position optimization.
- Dock and facility-capacity utilization.
- Reduction of unnecessary transfers and handling operations.
- Preservation of network resilience under disruption.

These objectives may conflict. For example, transportation consolidation can reduce cost but potentially increase delivery time. Conversely, prioritizing expedited transportation can improve responsiveness while increasing operational expense. A practical scheduling architecture must therefore support configurable objective priorities, constraints, and business policies.

F. Constraint Interdependencies

A defining characteristic of cross-network scheduling is the interdependency among operational constraints. Individual constraints cannot always be evaluated independently because the feasibility of one activity may depend on the availability of resources at another node.

For example, inventory availability alone does not guarantee fulfillment feasibility. The associated warehouse must have sufficient processing capacity, the shipment must have transportation capacity, and the receiving location must have an appropriate delivery window and available receiving capacity.

The relationship can be represented as:
Inventory Availability + Processing Capacity + Transportation Capacity + Time Window + Resource Availability → Feasible Schedule
This dependency structure requires optimization models to consider multiple constraint classes simultaneously rather than treating each functional system as an independent decision domain.

G. Event-Driven Scheduling Architecture

Event-driven integration provides a mechanism for connecting operational changes with scheduling decisions. Enterprise systems can publish events representing changes in orders, inventory, transportation, facility capacity, or delivery commitments. These events can be evaluated by the scheduling layer to determine whether existing schedules remain feasible.

| ISSN: 2349-6037 | www.ijmserh.com ||Impact Factor: 9.274||Quarterly, Peer Reviewed & Refereed Journal|
|| Volume 12, Issue 3, July-September 2024 ||
DOI: 10.15662/IJMSERH.2024.1203049

Representative operational events include:

- OrderCreated
- InventoryUpdated
- ShipmentDelayed
- CarrierCapacityChanged
- FacilityCapacityChanged
- DeliveryWindowChanged
- OrderPriorityChanged
- ShipmentCompleted

An event-driven architecture enables selective reoptimization. For example, a transportation delay can trigger analysis of only the shipments, facilities, and customer commitments dependent on the affected transportation segment. If the impact remains localized, the scheduling system can modify the affected portion of the schedule without unnecessarily recalculating unrelated activities.

H. Human-in-the-Loop Decision Management

Although optimization can automate a significant portion of scheduling activities, operational environments may require human intervention for exceptional or high-impact decisions. Contractual obligations, strategic customers, unusual disruptions, and business-policy exceptions may require planner review.

A unified scheduling framework can therefore support multiple decision modes, including:
Automated Optimization → Planner Review → Approved Execution
Routine decisions satisfying predefined policies can be executed automatically, while exceptions can be routed to authorized planners. Human users may also override optimization recommendations when additional business information is unavailable to the scheduling model.
This approach combines computational scalability with operational governance and provides greater transparency into automated scheduling decisions.

I. Progression Toward Adaptive Network Scheduling

The evolution of retail scheduling can be characterized as a progression from localized planning toward continuously adaptive network coordination:
Facility Scheduling → Functional Scheduling → Integrated Scheduling → Cross-Network Optimization → Adaptive Network Scheduling
In an adaptive network scheduling model, operational state, optimization models, execution systems, and monitoring mechanisms form a continuous feedback loop:
Network State → Optimization → Schedule Generation → Execution → Monitoring → Event Detection → Reoptimization

This progression establishes the conceptual foundation for unified retail distribution scheduling. The objective is not necessarily to replace existing warehouse, transportation, inventory, or enterprise systems, but to introduce an intelligent coordination layer capable of evaluating their interdependencies and generating network-aware scheduling decisions.

Such an architecture provides a scalable foundation for subsequent integration of predictive analytics, mathematical optimization, event-driven orchestration, and intelligent decision-support mechanisms. The following sections examine the architecture and technical components required to implement this unified scheduling model.

III. PROPOSED UNIFIED CROSS-NETWORK SCHEDULING ARCHITECTURE

A unified cross-network scheduling architecture provides a coordination layer between distributed retail operations and scheduling intelligence. The primary objective is to establish a common decision framework through which demand, inventory, transportation, facility capacity, operational constraints, and execution events can be evaluated collectively. Rather than replacing existing enterprise applications, the architecture integrates heterogeneous operational systems and provides network-level scheduling capabilities across their existing boundaries.

International Journal of Multidisciplinary and Scientific Emerging ResearcH (IJMSERH)

| ISSN: 2349-6037 | www.ijmserh.com ||Impact Factor: 9.274||Quarterly, Peer Reviewed & Refereed Journal|

|| Volume 12, Issue 3, July-September 2024 ||

DOI: 10.15662/IJMSERH.2024.1203049

Fig 1: Cross-Network Scheduling Architecture Diagram

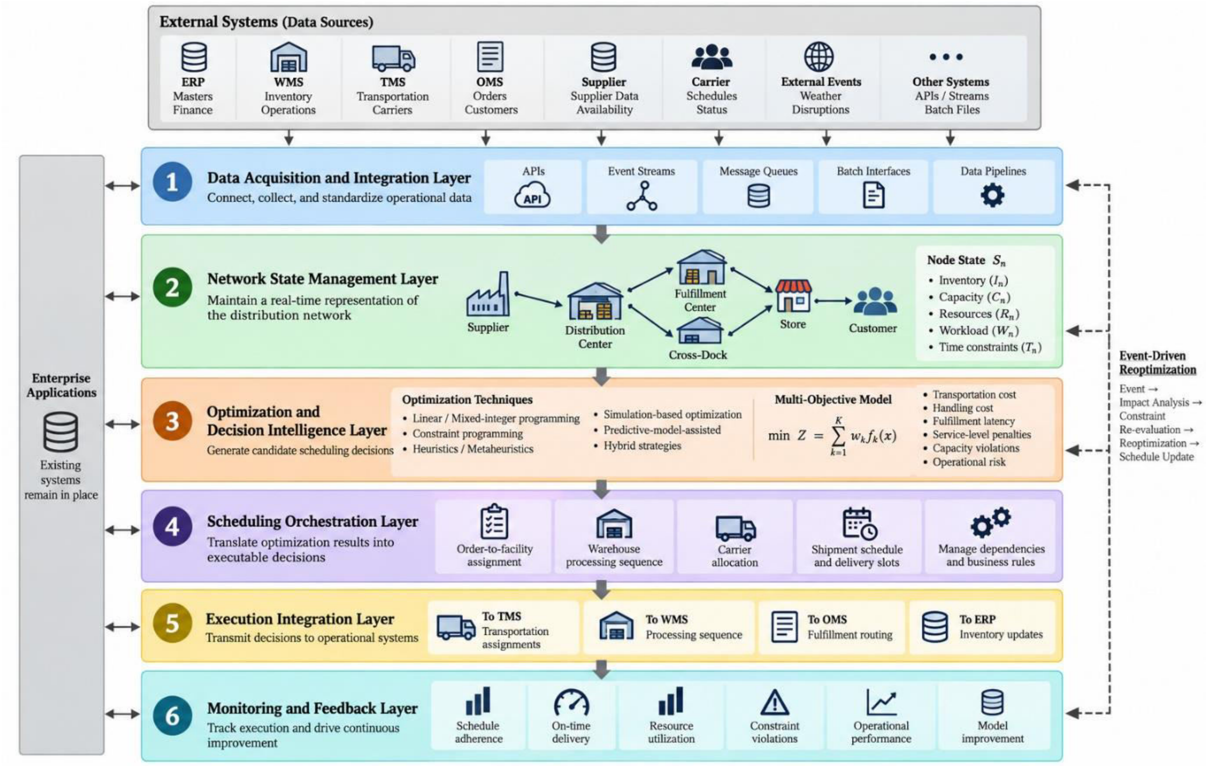

Fig. 1. Proposed unified cross-network scheduling architecture.

The proposed architecture can be organized into six logical layers: data acquisition, network state management, intelligence and optimization, scheduling orchestration, execution integration, and monitoring and feedback. This layered structure separates operational data collection from optimization logic and execution while providing well-defined interfaces between the components.

A. Data Acquisition and Integration Layer

The data acquisition layer provides connectivity between the scheduling platform and existing enterprise systems. Typical source systems include enterprise resource planning (ERP), warehouse management systems (WMS), transportation management systems (TMS), order management systems (OMS), inventory platforms, carrier systems, supplier interfaces, and external event sources.

The layer can support multiple integration mechanisms, including application programming interfaces (APIs), event streams, message queues, batch interfaces, and managed data pipelines. The objective is to transform heterogeneous operational information into standardized scheduling inputs.

Representative data elements include:

- Customer orders and delivery commitments.
- Inventory quantities and locations.
- Warehouse processing capacity.
- Labor and equipment availability.
- Transportation capacity.
- Carrier schedules.
- Delivery and pickup time windows.
- Facility operating calendars.
- Supplier availability.
- Shipment status.
- Historical operational performance.

| ISSN: 2349-6037 | www.ijmserh.com ||Impact Factor: 9.274||Quarterly, Peer Reviewed & Refereed Journal|

|| Volume 12, Issue 3, July-September 2024 ||

DOI: 10.15662/IJMSERH.2024.1203049

A standardized data contract is particularly important because source systems may use different identifiers, timestamps, units of measure, and data structures. Normalization at the integration boundary allows the optimization layer to operate on a consistent representation of network state.

B. Network State Management Layer

The network state layer maintains a current representation of the distribution environment. It combines relatively static information, such as facility locations and transportation lanes, with dynamic information, such as inventory levels, order status, available capacity, and shipment conditions.

The network can be represented as a directed graph:

$$G = (V, E)$$

where V represents distribution nodes and E represents transportation or operational relationships between those nodes.

A node may represent a distribution center, fulfillment center, store, supplier, cross-dock facility, or other operational location. An edge may represent a transportation lane, inventory transfer relationship, or fulfillment connection.

For each node $v \in V$, the state may include:

$$S_v = \{I_v, C_v, R_v, W_v, T_v\}$$

where:

- I_v = inventory state,
- C_v = processing capacity,
- R_v = available resources,
- W_v = workload,
- T_v = applicable time constraints.

Similarly, transportation edges can maintain information regarding available capacity, transit time, cost, carrier assignment, and operating windows.

This representation allows the scheduling engine to evaluate both node-level and network-level constraints.

C. Optimization and Decision Intelligence Layer

The optimization layer represents the analytical core of the architecture. It receives the current network state together with demand requirements, business priorities, and operational constraints and generates candidate scheduling decisions.

Multiple optimization techniques can be supported depending on network characteristics. These may include:

- Linear programming.
- Mixed-integer optimization.
- Constraint programming.
- Heuristic optimization.
- Metaheuristic approaches.
- Simulation-based optimization.
- Predictive-model-assisted optimization.
- Hybrid optimization strategies.

The optimization layer does not necessarily need to produce a single globally optimal solution for every operational scenario. In large-scale networks, computational complexity may make exact optimization impractical within operational time constraints. Consequently, the architecture can support configurable solution strategies that balance optimization quality with execution latency.

D. Multi-Objective Scheduling Model

The scheduling engine can evaluate several objectives simultaneously. A generalized objective function can be represented as:

$$\min Z = w_1 C_{transport} + w_2 C_{handling} + w_3 L_{fulfillment} + w_4 P_{service} + w_5 P_{capacity} + w_6 P_{risk}$$

where:

- $C_{transport}$ represents transportation cost,
- $C_{handling}$ represents handling or transfer cost,
- $L_{fulfillment}$ represents fulfillment latency,
- $P_{service}$ represents service-level penalties,
- $P_{capacity}$ represents capacity violation penalties,

| ISSN: 2349-6037 | www.ijmserh.com ||Impact Factor: 9.274||Quarterly, Peer Reviewed & Refereed Journal|

|| Volume 12, Issue 3, July-September 2024 ||

DOI: 10.15662/IJMSERH.2024.1203049

- P_{risk} represents operational risk,
- $w_1, \ldots, w_6$ represent configurable objective weights.

The formulation can be extended according to the requirements of a specific distribution network. Hard constraints can be used for requirements that must never be violated, while soft constraints can be represented through penalties or priority weights.

E. Scheduling Orchestration Layer

The orchestration layer converts optimization results into executable scheduling decisions. It determines how recommendations should be distributed to downstream operational systems.

A scheduling decision may include:

- Order-to-facility assignment.
- Warehouse processing sequence.
- Shipment consolidation.
- Carrier allocation.
- Transportation departure time.
- Dock assignment.
- Inventory transfer.
- Fulfillment-path selection.
- Delivery-slot assignment.

The orchestration layer also manages dependencies between decisions. For example, a shipment should not be released before the associated warehouse processing activity is completed or before the required inventory has been confirmed.

This layer therefore acts as a bridge between optimization and execution, ensuring that scheduling decisions are translated into operationally valid actions.

F. Event-Driven Reoptimization

A key capability of the architecture is event-driven reoptimization. Operational events can continuously modify the state of the network.

A generalized process can be expressed as:

Event → Impact Analysis → Constraint Re-evaluation → Reoptimization → Schedule Update

For example, if a transportation provider reports a delay, the system can identify the affected shipments, downstream facilities, and customer commitments. The optimization engine can then evaluate alternative transportation or fulfillment paths.

This selective approach prevents unnecessary recalculation of unaffected portions of the network.

G. Execution Integration Layer

The execution layer connects the scheduling architecture to operational systems. Depending on enterprise requirements, scheduling decisions can be transmitted through APIs, messages, events, or transactional interfaces.

For example:

Scheduling Engine → TMS: transportation assignment and shipment schedule.

Scheduling Engine → WMS: warehouse processing sequence and dispatch requirements.

Scheduling Engine → OMS: fulfillment-location or order-routing decision.

Scheduling Engine → ERP: inventory movement or operational planning updates.

The separation between scheduling intelligence and execution systems allows existing applications to continue performing their specialized transactional functions while the unified scheduling layer coordinates decisions across them.

H. Monitoring and Feedback Layer

The final architectural layer provides continuous monitoring of schedule execution. Planned activities are compared with actual operational outcomes to identify deviations and performance gaps.

Relevant measurements may include:

- Schedule adherence.
- Order fulfillment latency.
- Transportation utilization.
- Warehouse utilization.
- On-time delivery.
- Inventory transfer frequency.
- Schedule revision frequency.
- Constraint violations.

International Journal of Multidisciplinary and Scientific Emerging ResearcH (IJMSERH)

| ISSN: 2349-6037 | www.ijmserh.com ||Impact Factor: 9.274||Quarterly, Peer Reviewed & Refereed Journal|

|| Volume 12, Issue 3, July-September 2024 ||

DOI: 10.15662/IJMSERH.2024.1203049

- Resource utilization.
- Optimization execution time.

These measurements can be fed back into the scheduling process to support continuous improvement.
For example, repeated deviations between planned and actual warehouse processing times may indicate that the scheduling model requires revised processing-time estimates. Similarly, recurring transportation delays on a particular lane may require additional risk factors or alternative routing options.

I. Closed-Loop Scheduling Architecture

The complete architecture therefore operates as a closed-loop system:
Data Acquisition → Network State → Optimization → Scheduling → Execution → Monitoring → Feedback → Reoptimization
This closed-loop structure distinguishes cross-network scheduling from conventional static planning. The schedule is treated as a continuously evaluated operational artifact rather than a fixed plan generated once for an entire planning period.

The architecture also supports incremental implementation. Organizations can initially integrate a limited number of facilities or transportation lanes and subsequently expand the scheduling domain. Existing transactional applications can remain in place while optimization and orchestration capabilities are introduced progressively.

Such an architecture provides the technical foundation for coordinating distributed retail operations while preserving integration with established enterprise systems. The next section can formalize the mathematical optimization model and constraints for cross-network scheduling, including decision variables, objective functions, capacity constraints, inventory constraints, transportation constraints, and service-level requirements.

IV. MATHEMATICAL MODEL FOR CROSS-NETWORK SCHEDULING OPTIMIZATION

Cross-network scheduling can be formulated as a constrained multi-objective optimization problem in which orders, inventory, transportation resources, facilities, and time windows are coordinated across a distributed retail network. The mathematical model provides a formal mechanism for converting operational requirements into decision variables, objective functions, and constraints. Although the exact formulation varies according to network characteristics, a generalized model can provide a common foundation for implementation across different retail environments.

Fig 2 : Cross-Network Scheduling Optimization Model

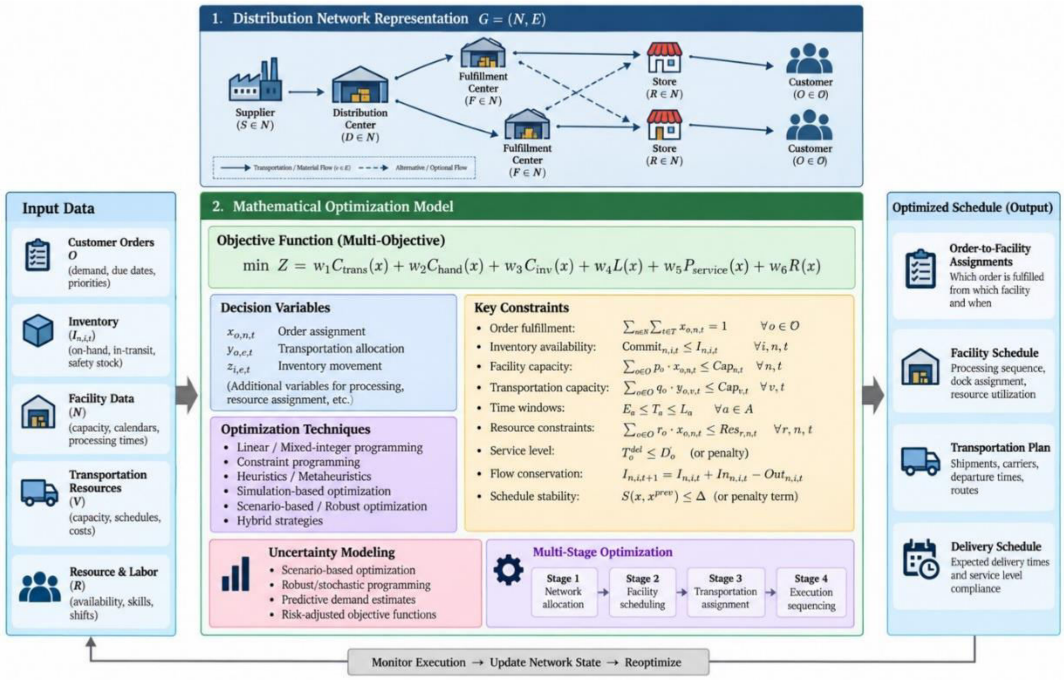

Fig. 2. Mathematical model for cross-network scheduling optimization.

International Journal of Multidisciplinary and Scientific Emerging ResearcH (IJMSERH)

| ISSN: 2349-6037 | www.ijmserh.com ||Impact Factor: 9.274||Quarterly, Peer Reviewed & Refereed Journal|

|| Volume 12, Issue 3, July-September 2024 ||

DOI: 10.15662/IJMSERH.2024.1203049

A. Network Representation

Let the distribution network be represented as a directed graph:

$$G = (V, E)$$

where V represents operational nodes and E represents transportation or material-flow connections.
The set of nodes can be defined as:

$$V = V_D \cup V_F \cup V_S \cup V_C$$

where:

- V_D = distribution centers,
- V_F = fulfillment centers,
- V_S = retail stores,
- V_C = other customer-facing or consolidation nodes.

The edge set E represents feasible transportation or transfer relationships between these nodes. Each edge may contain attributes such as transportation cost, expected transit time, available capacity, carrier availability, and operating restrictions.
Orders are represented by a set O, transportation resources by K, and planning time periods by T.

B. Decision Variables

The scheduling model requires variables that represent the principal operational decisions.
Let:

$$x_{o,v,t} = \begin{cases} 1, & \text{if order } o \text{ is assigned to node } v \text{ at time } t \\ 0, & \text{otherwise} \end{cases}$$

This variable represents order-to-node and time assignment.
Transportation allocation can be represented as:

$$y_{e,k,t} = \begin{cases} 1, & \text{if transportation resource } k \text{ uses edge } e \text{ at time } t \\ 0, & \text{otherwise} \end{cases}$$

Inventory movement can be represented by:

$$q_{i,e,t}$$

where $q_{i,e,t}$ represents the quantity of item i transferred through edge e during time period t.
Processing decisions can similarly be represented through variables describing warehouse workload, dock utilization, or resource assignment.
These variables collectively allow the optimization engine to coordinate order fulfillment, inventory movement, facility processing, and transportation scheduling.

C. Objective Function

A practical retail scheduling problem generally requires multiple objectives. A generalized weighted objective function can be expressed as:

$$\min Z = w_1 C_T + w_2 C_H + w_3 C_I + w_4 L + w_5 P_S + w_6 P_R$$

where:

- C_T = transportation cost,
- C_H = handling and transfer cost,
- C_I = inventory movement cost,
- L = fulfillment and transportation latency,
- P_S = service-level penalty,
- P_R = operational risk or disruption penalty,
- w_i = configurable weights.

The weights allow organizations to represent different operating priorities without changing the underlying architecture. For example, a time-sensitive fulfillment environment may assign greater importance to fulfillment latency and service-level adherence, whereas a cost-sensitive replenishment operation may emphasize transportation and handling costs.

| ISSN: 2349-6037 | www.ijmserh.com ||Impact Factor: 9.274||Quarterly, Peer Reviewed & Refereed Journal|

|| Volume 12, Issue 3, July-September 2024 ||

DOI: 10.15662/IJMSERH.2024.1203049

D. Order Fulfillment Constraints

Each customer order should be assigned to an appropriate fulfillment path while satisfying order-level requirements.
A basic assignment constraint can be represented as:

$$\sum_{v \in V} \sum_{t \in T} x_{o,v,t} = 1 \; \forall o \in O$$

This ensures that each order receives one selected fulfillment assignment within the planning horizon.
Additional constraints can enforce customer delivery windows, order priorities, product availability, and permissible fulfillment locations.
For example, if order $o$must be delivered within a defined time interval, the scheduling engine must prevent assignments that result in delivery outside that interval.

E. Inventory Availability Constraints

A scheduling decision cannot allocate more inventory than is physically or operationally available.
For item i, node v, and time t, inventory feasibility can be represented as:

$$I_{i,v,t}^{available} - I_{i,v,t}^{allocated} \geq 0$$

where $I^{available}$represents available inventory and $I^{allocated}$represents the quantity committed to scheduled activities.
The model can further distinguish between on-hand inventory, reserved inventory, safety stock, in-transit inventory, and inventory that is unavailable because of quality or operational restrictions.
This distinction is particularly important in omnichannel environments where several demand channels may compete for the same inventory pool.

F. Facility Capacity Constraints

Distribution and fulfillment facilities have finite processing capacity. Let $C_{v,t}$represent the maximum processing capacity of node $v$during time period t.
The workload assigned to the node must satisfy:

$$\sum_{o \in O} p_{o,v,t} \, x_{o,v,t} \leq C_{v,t}$$

where $p_{o,v,t}$represents the processing requirement associated with order o.
This constraint can be extended to model different resource types, including labor, automation equipment, packing stations, docks, storage areas, and specialized processing capabilities.

G. Transportation Capacity Constraints

Transportation resources also impose constraints on the scheduling problem. If $Cap_{k,t}$represents the available capacity of transportation resource $k$during time period t, the assigned shipment volume must satisfy:

$$\sum_{o \in O} q_{o,k,t} \leq Cap_{k,t}$$

This constraint can incorporate weight, volume, pallet count, container capacity, or other operational measures.
Transportation availability may also vary according to carrier schedules, equipment type, geographic restrictions, and delivery windows.

H. Time-Window Constraints

Retail distribution operations commonly involve pickup, processing, and delivery time windows.
For each scheduled activity:

$$a_o \geq E_o$$

and

$$a_o \leq L_o$$

where:

- a_o= scheduled activity time,
- E_o= earliest permissible time,
- L_o= latest permissible time.

International Journal of Multidisciplinary and Scientific Emerging ResearcH (IJMSERH)

| ISSN: 2349-6037 | www.ijmserh.com ||Impact Factor: 9.274||Quarterly, Peer Reviewed & Refereed Journal|

|| Volume 12, Issue 3, July-September 2024 ||

DOI: 10.15662/IJMSERH.2024.1203049

These constraints prevent the scheduling engine from generating operationally infeasible schedules.
Time windows can also be applied to facility receiving operations, carrier availability, store receiving periods, and customer delivery commitments.

I. Resource and Workforce Constraints

Warehouse operations depend on the availability of human and automated resources. If $R_{v,r,t}$ represents the available quantity of resource r at node v during time t, the workload assigned to that resource cannot exceed its capacity.
A generalized constraint is:

$$\sum_{o \in O} u_{o,r,v,t} \, x_{o,v,t} \leq R_{v,r,t}$$

where $u_{o,r,v,t}$ represents the resource requirement of order o.
This allows the model to incorporate labor shifts, equipment availability, maintenance periods, and other operational restrictions.

J. Service-Level Constraints

Service-level requirements can be represented as hard constraints or penalty-based objectives.
For example, if D_o represents the required delivery time and A_o represents the actual or scheduled delivery time:

$$A_o \leq D_o$$

can be applied when the service commitment is mandatory.
Alternatively, late deliveries can be permitted but penalized:

$$P_S = \sum_{o \in O} \max \left(0, A_o - D_o\right)$$

This approach allows the optimization engine to evaluate the operational cost of service-level deviations while still generating feasible solutions when perfect compliance is impossible.

K. Network Flow Conservation

Cross-network scheduling must also maintain consistency of material movement. Inventory entering and leaving a node should satisfy flow-balance requirements.
A simplified formulation is:

$$I_{v,t+1} = I_{v,t} + In_{v,t} - Out_{v,t}$$

where:

- $I_{v,t}$ = inventory at node v and time t,
- $In_{v,t}$ = inbound inventory,
- $Out_{v,t}$ = outbound inventory.

This constraint prevents the optimization engine from generating schedules that depend on inventory that cannot physically reach the relevant facility within the required time.

L. Schedule Stability Constraints

Continuous reoptimization can create excessive schedule changes if every new event results in a complete restructuring of the existing plan. To address this problem, a schedule-stability penalty can be introduced.
Let x^{old} represent the previously committed schedule and x^{new} the proposed schedule. A stability term can be represented as:

$$C_{stability} = \sum \left| x^{new} - x^{old} \right|$$

The optimization objective can then include:

$$\min Z' = Z + w_s C_{stability}$$

where w_s controls the importance of schedule stability.
This mechanism encourages the optimizer to modify only those activities that require change while preserving already committed operational decisions whenever possible.

M. Multi-Stage Optimization

Large retail networks may contain thousands or millions of possible scheduling combinations. A single monolithic optimization model may therefore become computationally expensive.
A hierarchical approach can be used in which scheduling is performed in multiple stages:
Stage 1: Network-level demand and capacity allocation.
Stage 2: Facility-level workload scheduling.
Stage 3: Transportation assignment and consolidation.
Stage 4: Detailed execution sequencing.
This decomposition reduces computational complexity and allows specialized optimization models to operate at different levels of the network.

N. Optimization Under Uncertainty

Retail scheduling involves uncertainty in demand, transportation duration, processing time, and resource availability. Consequently, deterministic optimization may not always provide sufficient protection against operational variability.
The model can incorporate uncertainty through:

- Scenario-based optimization.
- Robust optimization.
- Stochastic programming.
- Predictive demand estimates.
- Safety-capacity buffers.
- Risk-adjusted objective functions.

For example, transportation time can be represented using expected and worst-case estimates rather than a single fixed value. This allows schedules to incorporate operational risk rather than optimizing solely against nominal conditions.

O. Generalized Optimization Workflow

The mathematical model can be integrated into an operational scheduling workflow:
Demand and Network State → Decision Variables → Constraint Evaluation → Multi-Objective Optimization → Feasibility Check → Schedule Generation → Execution
The resulting schedule can then be monitored against actual execution data. Deviations can trigger incremental reoptimization, creating a closed-loop relationship between the mathematical model and operational execution.

V. DATA ARCHITECTURE AND REAL-TIME EVENT INTEGRATION

The effectiveness of cross-network scheduling depends on the availability, consistency, timeliness, and quality of operational data. A scheduling engine can generate theoretically optimal decisions only when the underlying representation of network conditions is sufficiently accurate. In distributed retail environments, however, operational information is typically fragmented across warehouse management systems, transportation platforms, enterprise resource planning systems, order management applications, carrier interfaces, inventory repositories, and external data sources.
A unified data architecture is therefore required to transform heterogeneous operational information into a consistent and continuously updated representation of network state. This architecture should support both historical data required for analytics and real-time events required for dynamic scheduling.

A. Distributed Data Sources

A retail distribution network may generate scheduling-relevant information from numerous sources. Representative systems include:

- Enterprise resource planning (ERP) platforms.
- Warehouse management systems (WMS).
- Transportation management systems (TMS).
- Order management systems (OMS).
- Inventory management platforms.
- Supplier and carrier systems.
- Store systems.
- Customer-order channels.
- IoT and warehouse sensors.
- External transportation and environmental data sources.

| ISSN: 2349-6037 | www.ijmserh.com ||Impact Factor: 9.274||Quarterly, Peer Reviewed & Refereed Journal|

|| Volume 12, Issue 3, July-September 2024 ||

DOI: 10.15662/IJMSERH.2024.1203049

Each system may expose information using different data structures, identifiers, interfaces, and update frequencies. For example, an order management system may provide order-level requirements, while a warehouse management system provides processing status and facility capacity. A transportation system may provide carrier availability and estimated arrival information.

The data architecture must therefore establish a common integration mechanism without requiring all operational systems to adopt identical technologies.

B. Canonical Network Data Model

A canonical data model can provide a standardized representation of scheduling entities across the network. Core entities may include:

Entity	Representative Attributes
Order	Order ID, priority, quantity, destination, commitment time
Product	SKU, category, dimensions, handling requirements
Inventory	Location, available quantity, reserved quantity, status
Facility	Location, capacity, operating calendar, resources
Shipment	Shipment ID, origin, destination, carrier, status
Transportation Resource	Vehicle type, capacity, availability
Delivery Window	Start time, end time, destination
Resource	Labor, equipment, dock, processing capability
Event	Event type, timestamp, source, affected entity

A canonical representation reduces the dependency of the optimization layer on individual source-system data models. It also enables scheduling algorithms to operate consistently across heterogeneous network environments.

C. Data Ingestion Architecture

Data can enter the scheduling platform through both batch and real-time mechanisms. Batch pipelines remain useful for relatively stable information such as historical order data, facility calendars, reference information, and periodic forecasts. Real-time integration is more appropriate for rapidly changing operational information.

A generalized ingestion pattern is:

Source Systems → Integration Layer → Validation → Transformation → Network State → Optimization Engine

The integration layer can use APIs, event streaming, message brokers, managed data pipelines, or other enterprise integration mechanisms.

Data validation should occur before information is incorporated into the network state. Invalid timestamps, duplicate events, inconsistent identifiers, negative quantities, or unavailable facility references can otherwise propagate erroneous conditions into the scheduling model.

D. Event-Driven Integration

Event-driven integration provides the foundation for responsive scheduling. Instead of periodically polling every operational system, relevant applications can publish events when important state changes occur.

Representative events include:

Order Events
- Order created.
- Order modified.
- Order cancelled.
- Order priority changed.

Inventory Events
- Inventory received.
- Inventory allocated.
- Inventory transferred.
- Inventory shortage detected.

Transportation Events
- Shipment created.

| ISSN: 2349-6037 | www.ijmserh.com ||Impact Factor: 9.274||Quarterly, Peer Reviewed & Refereed Journal|
|| Volume 12, Issue 3, July-September 2024 ||
DOI: 10.15662/IJMSERH.2024.1203049

- Vehicle assigned.
- Shipment delayed.
- Carrier capacity changed.
- Shipment delivered.

Facility Events

- Capacity reduced.
- Equipment unavailable.
- Processing backlog detected.
- Facility temporarily restricted.

Each event should contain sufficient metadata to identify the affected entity, event time, source system, and relevant state change.

E. Event Processing and Impact Analysis

Not every operational event requires complete network reoptimization. A critical capability of the architecture is determining the scope of an event's impact.

A generalized process can be expressed as:

Event Detection → Entity Identification → Dependency Analysis → Impact Estimation → Constraint Re-evaluation → Selective Reoptimization

For example, a delay affecting a single transportation lane may influence a subset of shipments and downstream facilities. The scheduling system can identify those dependencies and determine whether alternative transportation, inventory allocation, or fulfillment paths are required.

This approach reduces unnecessary computational workloads and minimizes disruption to unaffected schedules.

F. Real-Time Network State

The scheduling platform should maintain a continuously updated operational state rather than relying exclusively on periodic snapshots.

A simplified network-state representation can be defined as:

$$S(t) = \{O,(t), I,(t), C,(t)T(t)R(t)E(t)\}$$

where:

- $O(t)$= order state,
- $I(t)$= inventory state,
- $C(t)$= facility capacity state, l;
- $T(t)$= transportation state,
- $R(t)$= resource state,
- $E(t)$= active operational events.

At any point in time, the optimization engine can consume the current state $S(t)$to evaluate the feasibility and quality of the active schedule.

This model also enables historical state reconstruction, which can be useful for performance analysis and post-event investigation.

G. Data Quality and Consistency

Data quality is particularly important because optimization algorithms can amplify the consequences of incorrect inputs. A scheduling engine operating with inaccurate inventory levels or outdated transportation capacity may generate mathematically valid but operationally infeasible schedules.

A data-quality framework should therefore address:

- Completeness.
- Accuracy.
- Timeliness.
- Consistency.
- Uniqueness.
- Referential integrity.
- Timestamp synchronization.

Validation rules can be implemented at ingestion time, while anomaly-detection mechanisms can identify unexpected changes in operational data.

| ISSN: 2349-6037 | www.ijmserh.com ||Impact Factor: 9.274||Quarterly, Peer Reviewed & Refereed Journal|

|| Volume 12, Issue 3, July-September 2024 ||

DOI: 10.15662/IJMSERH.2024.1203049

For example, a sudden increase in available inventory that exceeds expected receiving capacity may require validation before being incorporated into an optimization run.

H. Event Ordering and Temporal Consistency

Distributed systems can produce events at different times and in different sequences. An inventory update may arrive after an order allocation event even though the physical operations occurred in the opposite order.

The event-processing architecture should therefore support temporal consistency through event timestamps, sequence identifiers, version numbers, or other mechanisms.

A simplified event representation can be expressed as:

$$E_i = (I, D_i, T, y, p, e_i Source_i Time_i Version_i Payload_i)$$

where the event identifier, event type, source, timestamp, version, and payload collectively support reliable event processing.

This becomes particularly important when multiple systems independently update the same business entity.

I. Integration with Optimization Services

Once data has been validated and incorporated into the network state, the optimization service can be invoked according to predefined triggers.

Optimization triggers may include:

- New high-priority orders.
- Significant inventory changes.
- Transportation disruptions.
- Facility-capacity changes.
- Service-level risk.
- Scheduled planning intervals.
- Threshold-based workload changes.

The architecture can distinguish between full optimization and incremental optimization. Full optimization evaluates a broad portion of the network, whereas incremental optimization focuses on the affected nodes, orders, resources, and transportation relationships.

J. Historical Data and Learning

Real-time scheduling should be complemented by historical data analysis. Historical execution records provide information about actual processing times, transportation delays, capacity utilization, order patterns, and schedule deviations.

This information can be used to improve future scheduling parameters.

For example:

$$Expected\ Processing\ Time = f(O, r, d, er\ TypeFacilityWorkloadHistorical\ Performance)$$

Similarly, expected transportation duration can incorporate historical lane-level performance rather than relying exclusively on nominal transit times.

Predictive models can subsequently provide estimates to the optimization engine, allowing scheduling decisions to reflect expected operating conditions.

K. Data Governance and Security

Because the scheduling architecture integrates information across multiple enterprise systems, data governance must be incorporated into the design. Access controls should ensure that users and services receive only the information required for their operational responsibilities.

Security controls can include:

- Identity-based authentication.
- Role-based authorization.
- API security.
- Encryption in transit and at rest.
- Data lineage.
- Audit logging.
- Event-level traceability.
- Retention and archival policies.

| ISSN: 2349-6037 | www.ijmserh.com ||Impact Factor: 9.274||Quarterly, Peer Reviewed & Refereed Journal|

|| Volume 12, Issue 3, July-September 2024 ||

DOI: 10.15662/IJMSERH.2024.1203049

Data governance also helps establish accountability for scheduling decisions by maintaining a traceable relationship between source information, optimization inputs, generated schedules, and execution outcomes.

L. Unified Event-to-Schedule Pipeline

The complete data and event integration process can therefore be represented as:
Operational Systems → Data Ingestion → Validation and Normalization → Unified Network State → Event Processing → Impact Analysis → Optimization → Schedule Update → Execution Feedback
This architecture transforms fragmented operational information into a continuously updated scheduling context. It also provides the foundation for incorporating predictive intelligence and machine-learning models into the scheduling process.

VIII. IMPLEMENTATION CONSIDERATIONS

Implementation of cross-network scheduling should generally follow an incremental approach. A complete replacement of existing operational systems is neither necessary nor desirable in many enterprise environments.
A practical implementation sequence can include:
Phase 1: Establish data integration and network visibility.
Phase 2: Introduce unified network-state management.
Phase 3: Implement constraint-based scheduling.
Phase 4: Introduce predictive models.
Phase 5: Enable event-driven reoptimization.
Phase 6: Introduce digital-twin or simulation-based evaluation.
Phase 7: Expand toward continuous adaptive scheduling.
This incremental model allows organizations to validate individual capabilities before expanding the scheduling domain.
Integration architecture should also support interoperability with existing WMS, TMS, OMS, ERP, and analytics platforms. APIs, event-driven interfaces, standardized data contracts, and asynchronous messaging can reduce coupling between the optimization platform and operational applications.
Security, observability, model governance, auditability, and failure handling should be treated as architectural requirements rather than post-implementation enhancements.

IX. CONCLUSION

Cross-network scheduling optimization provides a framework for coordinating increasingly interconnected retail distribution operations across facilities, transportation networks, inventory locations, fulfillment channels, and shared resources. Traditional facility-level and function-specific scheduling approaches can provide effective local decisions but may fail to account for dependencies that extend across the broader distribution network.

The architecture presented in this article addresses this limitation through a unified scheduling model that combines network-state management, mathematical optimization, event-driven integration, predictive analytics, and continuous feedback. The proposed model represents distribution operations as an interconnected network in which inventory, capacity, transportation, time windows, resources, and service commitments are evaluated collectively.

A mathematical formulation establishes explicit decision variables and constraints for order assignment, inventory availability, facility capacity, transportation capacity, time windows, resource utilization, service-level adherence, network flow, and schedule stability. This formulation provides a technology-neutral foundation that can be implemented using different optimization techniques according to network size and operational requirements.

The integration of machine learning and predictive analytics further extends the capabilities of the scheduling framework. Demand forecasts, transportation-time predictions, facility workload estimates, and risk indicators can be incorporated into optimization models to improve the representation of uncertain operating conditions. Digital twins and simulation environments can additionally provide mechanisms for evaluating alternative schedules before operational execution.

An important characteristic of the proposed approach is its closed-loop architecture. Operational events continuously update the network state, affected decisions can be re-evaluated, and revised schedules can be generated when required. This transforms scheduling from a static planning activity into an adaptive operational capability.

| ISSN: 2349-6037 | www.ijmserh.com ||Impact Factor: 9.274||Quarterly, Peer Reviewed & Refereed Journal|

|| Volume 12, Issue 3, July-September 2024 ||

DOI: 10.15662/IJMSERH.2024.1203049

The approach also emphasizes incremental enterprise adoption. Existing WMS, TMS, OMS, ERP, and transportation platforms can continue performing their transactional responsibilities while a unified scheduling layer coordinates decisions across system boundaries. This reduces the dependency on large-scale application replacement and enables organizations to progressively introduce optimization and intelligent decision support.

Ultimately, cross-network scheduling should not be viewed solely as an optimization algorithm. It represents an integrated architectural capability combining data, optimization, prediction, orchestration, execution, monitoring, and feedback. When these components operate as a coordinated system, retail distribution networks can establish a more responsive foundation for managing demand variability, shared capacity, transportation uncertainty, and operational disruptions.

Future research can extend this framework through autonomous multi-agent scheduling, explainable AI, reinforcement learning, carbon-aware transportation optimization, probabilistic digital twins, and collaborative optimization across retailers, carriers, suppliers, and logistics partners.

REFERENCES

The following references span 2020–2024 and are selected to support the article's discussion of omnichannel retail, digital supply chains, AI/ML, digital twins, optimization, resilience, and retail-network scheduling.

[1] B. Ageron, O. Bentahar, and A. Gunasekaran, "Digital supply chain: Challenges and future directions," *Supply Chain Forum: An International Journal*, vol. 21, no. 3, pp. 133–138, 2020.

[2] B. J. Gibson, R. Ishfaq, and B. Davis-Sramek, *2020 Annual State of the Retail Supply Chain Report*, Auburn University Center for Supply Chain Innovation, Retail Industry Leaders Association, and DC Velocity, 2020.

[3] A. Grewal, M. Noble, and others, "Examining retail business model transformation: A longitudinal study of the transition to omnichannel order fulfillment," *International Journal of Physical Distribution & Logistics Management*, vol. 50, no. 5, pp. 557–576, 2020.

[4] M. M. Pereira and E. M. Frazzon, "A data-driven approach to adaptive synchronization of demand and supply in omni-channel retail supply chains," *International Journal of Information Management*, vol. 57, Art. no. 102165, 2021.

[5] R. Ishfaq, "Digital supply chains in omnichannel retail: A conceptual framework," *Journal of Business Logistics*, vol. 43, no. 2, pp. 169–188, 2022.

[6] S. Islam, S. H. Amin, and L. J. Wardley, "Machine learning and optimization models for supplier selection and order allocation planning," *International Journal of Production Economics*, vol. 242, Art. no. 108315, 2021.

[7] L. Wang, T. Deng, Z.-J. M. Shen, H. Hu, and Y. Qi, "Digital twin-driven smart supply chain," *Frontiers of Engineering Management*, vol. 9, pp. 56–70, 2022.

[8] H. D. Perez, J. M. Wassick, and I. E. Grossmann, "A digital twin framework for online optimization of supply chain business processes," *Computers & Chemical Engineering*, vol. 166, Art. no. 107972, 2022.

[9] M. Andrejić, "Modeling retail supply chain efficiency: Exploration and comparative analysis of different approaches," *Mathematics*, vol. 11, no. 7, Art. no. 1571, 2023.

[10] J. Wang, C. L. E. Swartz, and K. Huang, "Deep learning-based model predictive control for real-time supply chain optimization," *Journal of Process Control*, vol. 129, Art. no. 103049, 2023.

[11] I. Alsolbi, F. H. Shavaki, R. Agarwal, G. K. Bharathy, S. Prakash, and M. Prasad, "Big data optimisation and management in supply chain management: A systematic literature review," *Artificial Intelligence Review*, vol. 56, pp. 253–284, 2023.

International Journal of Multidisciplinary and Scientific Emerging Research (IJMSERH)

Impact Factor: 9.274

ijmserh@gmail.com

www.ijmserh.com