

Original Article

Coordinating Supply Chain Agents Under Shared Inventory and Capacity Constraints

SACHIDANANDA SINGH

Senior Engineering Manager, Hims & Hers, USA.

ABSTRACT: Software agents acting for stores, a Distribution Centre (DC) and a plant share resources, and when their orders exceed supply, the rule that resolves the conflict shapes cost and service. This paper formulates coordination under four shared resources, namely DC stock, outbound transport, DC storage space and production capacity and proposes a coordination layer that aggregates agent state, detects conflicts, allocates scarce resources to equalise projected stock-out risk, and checks planned production against storage space. The layer is evaluated in a three-stage simulation driven by the real daily sales of 50 products per sample in ten Walmart stores (M5 data), over 20 samples, tuned on 2014 and tested on 536 later days. Proportional rationing cost 18.6% less than serving agents first come, first served. Echelon ordering on shared sales data cost 8.6% more, mainly because stock stranded at one store could not serve another. With stores merged, it cost 4.4% less. The coordination layer matched proportional rationing when capacity was slack and cost 4.7% to 11.9% less when production or transport ran at 95% to 98% utilisation, and it kept the fill rate 1.7 points higher during a production disruption. Balanced allocation drove most of these gains. The storage check helped only when transport was the bottleneck and raised cost when storage was tight, so balanced allocation without it was the best configuration in all other scenarios. Conflict-aware allocation pays off where resources run near their limits, and hard constraint checks must weigh the cost of what they prevent.

KEYWORDS: Multi-Agent Systems, Supply Chain Coordination, Shared Inventory, Capacity Constraints, Resource Allocation, Information Sharing, Bullwhip Effect, Inventory Management, M5 Dataset.

1. INTRODUCTION

A supply chain is run by many decision makers at once. A plant schedules production, a Distribution Centre (DC) replenishes its stock and ships to stores, and each store orders from the DC. When these decisions are delegated to software agents, each agent sees its own state and acts on its own objective, which suits organisations where control and information are spread across sites and firms [1], [2]. The difficulty is that the agents draw on the same limited resources. Several stores request the same DC stock, all products compete for the same production line, and all shipments leave through the same outbound dock and trucks. When requests exceed what is available, the rule that resolves the conflict decides who is served, and the orders that agents place in anticipation of being rationed can amplify variability upstream [3], [4].

The classical literature gives most of the ingredients for handling this. Sharing point-of-sale demand and inventory positions reduces the bullwhip effect [3], [5], [6]; echelon stock policies coordinate replenishment across stages [7], [8]; and allocation rules for scarce stock and capacity have been studied analytically [9], [10], [11], [12]. Agent-based studies of supply chains have proposed architectures in which a coordination layer collects state, detects conflicts and allocates resources [13], [14], [15], [16], and recent work has returned to autonomous agents built on reinforcement learning [17], [18], [19] and large language models [20], [21], [22]. Many of these agent studies describe architectures or report results in stylised settings, and few measure which parts of a coordination layer matter when several shared resources bind at once under real demand.

This paper studies that question in a controlled simulation. We model a three-stage chain with one plant, one DC and ten retail stores, in which the agents share four resources, namely DC stock, outbound transport capacity, DC storage space and plant production capacity. Customer demand is the real daily unit sales of 50 products in ten Walmart stores from the M5 dataset [23]. The agents themselves follow classical order-up-to policies, and the coordination rules we compare are built from classical components, namely proportional rationing, echelon ordering, balanced allocation and a storage check. The paper is therefore an empirical comparison of how these rules perform, alone and combined in a coordination layer, when several shared resources bind at once under real demand. It makes three contributions.

- A formulation of the shared-resource coordination problem with explicit inventory, transport, storage and production constraints, and a coordination layer that aggregates agent state, detects conflicts, allocates scarce stock, transport and production to equalise projected stock-out risk, and checks planned production against storage space (Sections 3 and 4).
- A comparison of five mechanisms (independent agents, proportional rationing, information sharing with echelon ordering, and the coordination layer with installation or echelon ordering), each with safety factors tuned on a separate calibration year, over 20 random product samples (Sections 5 and 6).

- An assessment of when coordination matters, across levels of production, transport and storage tightness, a production disruption, a regional demand surge and different cost parameters, with an ablation of the layer's components and measurements of bullwhip, communication and computation (Section 6).

2. RELATED WORK

2.1. MULTI-AGENT SYSTEMS FOR SUPPLY CHAINS

Agent-based models of supply chains date back more than two decades. Swaminathan et al. [13] represented supply chain entities as agents to study supply chain dynamics, and Fox et al. [14] proposed an agent-oriented architecture for coordinating supply chain functions. Lee et al. [24] applied a multi-agent system to supply chain management in a similar spirit. Later systems combined agents with case-based reasoning [1], collaborative inventory management [25], radio-frequency identification data [26] and big data processing [27], and autonomous trading agents were developed for supply chain competitions [28]. Surana et al. [2] described supply networks as complex adaptive systems whose behaviour emerges from the interaction of autonomous entities, and Herrera et al. [29] reviewed multi-agent systems in systems engineering. Xu et al. [16] discussed how autonomous supply chains can be implemented with multi-agent systems and identified coordination, interoperability and trust as the main open problems. Contract-net style negotiation [15] and case-based collaboration [1] are common coordination mechanisms in this literature. Coordination mechanisms more broadly, including contracts and information sharing, are reviewed by Li and Wang [30], and game-theoretic coordination by Hennet and Arda [31].

2.2. INFORMATION SHARING AND THE BULLWHIP EFFECT

Lee et al. [3] identified demand signal processing, rationing games, order batching and price variation as causes of the bullwhip effect, and Chen et al. [5] quantified how forecasting and lead times amplify order variance in a simple chain. Sterman [4] showed the same amplification in the beer game with human decision makers. Sharing point-of-sale data and inventory positions reduces this amplification, and its value has been quantified for uncapacitated [6] and capacitated [32] chains. Echelon stock policies, which base each stage's order on all stock downstream of it, are optimal for serial systems [7] and are a standard way to use shared information [8], [33]. Özer [34] reviews how information, coordination and incentives interact in inventory management.

2.3. ALLOCATION OF SCARCE INVENTORY AND CAPACITY

When a DC cannot meet all store orders, it must ration its stock. Eppen and Schrage [9] introduced allocation rules that balance the stores' stock-out risk, and van der Heijden et al. [10] compared balanced and proportional allocation in multi-echelon distribution systems. Cachon and Lariviere [11], [12] showed that proportional allocation of scarce capacity invites retailers to inflate their orders, which links rationing to the bullwhip effect. Coordinated replenishment across several retailers can lower cost compared with independent ordering [35], and collaborative planning schemes such as Collaborative Planning, Forecasting and Replenishment (CPFR) formalise the sharing of plans between partners [36], [37]. Safety stock placement across stages is treated by Graves and Willems [38]. Integrated planning under capacity limits is surveyed by Dias and Ierapetritou [39].

2.4. LEARNING-BASED AND LANGUAGE-MODEL AGENTS

Reinforcement learning has been applied to inventory control, production and transport decisions [17], and deep Q-networks have been trained to play the beer game [18]. Multi-agent reinforcement learning methods such as QMIX [40], MADDPG [41] and COMA [42] learn decentralised policies with centralised training, and Yang et al. [19] combined a demand forecaster with multi-agent reinforcement learning for retail inventory. Large language models have been used to explain supply chain optimisation results [20], to act as zero-shot inventory agents at each stage of a chain [21], and have been shown to display human-like biases, including bullwhip behaviour, when acting as inventory managers [22]. Language models can also produce fluent but unsupported output [43], which is one argument for checking their decisions against explicit constraints. These approaches change how an agent decides. The question studied here is complementary, namely what a coordination layer should do with the requests of agents, however those requests are produced.

3. PROBLEM FORMULATION

3.1. NETWORK AND AGENTS

The chain has one plant, one DC and a set R of retail stores, indexed by i , and carries a set P of products, indexed by p . Each store is represented by a store agent, the DC by a DC agent and the plant by a plant agent (Figure 1). Time is divided into days. Customer demand $d_{i,p,t}$ for product p at store i on day t is met from store stock, and demand that cannot be met is lost. Store orders that the DC cannot fill are backordered at the DC. Goods shipped by the DC are available at the store L_r days later, and production released by the plant is available at the DC L_m days later. Agents observe sales, not demand.

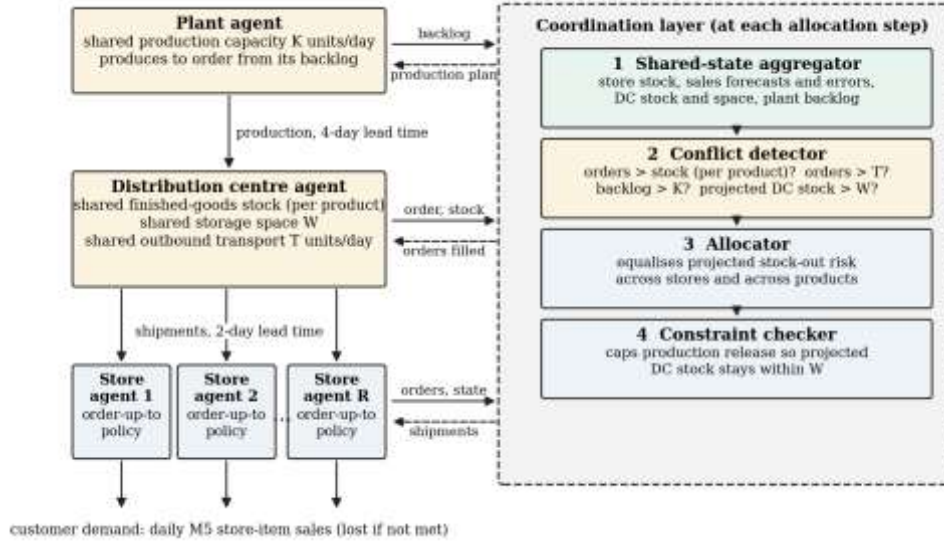


FIGURE 1 Supply Chain Network, Shared Resources and Coordination Layer. The Plant, The DC and the Stores are Agents. The Four Shared Resources are Production Capacity K , DC Stock, Outbound Transport Capacity T And Storage Space W . The Coordination Layer Collects Orders and State, Detects Conflicts, Allocates and Checks the Plan Before it is Executed

Each day proceeds in the same order (Figure 2). Production and shipments that are due arrive. Stores serve customer demand and update their forecasts from the day's sales. Stores place orders, the DC allocates its stock and transport to outstanding store orders, the DC orders from the plant, and the plant allocates its capacity to its backlog of DC orders. Costs are charged on end-of-day stock.

3.2. SHARED RESOURCES AND CONSTRAINTS

Let $u_{i,p,t}$ be the outstanding (backordered) orders of store i for product p at the DC after the store's order on day t , $x_{i,p,t}$ the quantity the DC ships, $J_{p,t}$ the DC's stock of product p before shipping, $B_{p,t}$ the plant's backlog of DC orders and $y_{p,t}$ the quantity the plant releases to production. The four shared resources impose the following constraints.

$$0 \leq x_{i,p,t} \leq u_{i,p,t} \quad \sum_i x_{i,p,t} \leq J_{p,t} \quad (\text{shared stock}) \quad (1)$$

$$\sum_i \sum_p x_{i,p,t} \leq T \quad (\text{shared outbound transport}) \quad (2)$$

$$0 \leq y_{p,t} \leq B_{p,t} \quad \sum_p y_{p,t} \leq K_t \quad (\text{shared production capacity}) \quad (3)$$

$$\sum_p J'_{p,t} \leq W \quad (\text{shared storage space, soft}) \quad (4)$$

Where K_t equals the nominal capacity K except during a disruption, and $J'_{p,t} = J_{p,t} - \sum_i x_{i,p,t}$ is end-of-day DC stock. Constraints (1) to (3) are hard. Constraint (4) is soft, because stock above W is kept in overflow space at a premium.

3.3. COSTS AND OBJECTIVE

With $I_{i,p,t}$ the store stock at the end of day t and $s_{i,p,t}$ the units sold, the daily cost is

$$C_t = h_r \sum_{i,p} I_{i,p,t} + h_d \sum_p J'_{p,t} + b \sum_{i,p} (d_{i,p,t} - s_{i,p,t}) + c_o [\sum_p J'_{p,t} - W]^+ \quad (5)$$

where h_r and h_d are holding costs per unit-day at the stores and the DC, b is the cost of a lost sale, and c_o is the overflow premium per unit-day. The objective of the chain is to minimise the mean daily cost over the horizon. Each agent controls only part of it, and the coordination mechanism decides how the agents' orders are reconciled with constraints (1) to (4).

4. COORDINATION FRAMEWORK

4.1. LOCAL AGENT POLICIES

All store agents use the same order-up-to policy, reviewed daily. Each keeps an exponentially weighted moving average of its daily sales, $\hat{\mu}_{i,p}$, and of its squared forecast error, $\hat{\sigma}_{i,p}^2$, with smoothing constant $\alpha = 0.1$. A unit ordered on day t can arrive at the earliest on day $t + L_r$, and the next order can arrive a day later, so the order must protect against demand over L_r days. The target and the order are

$$S_{i,p,t} = \hat{\mu}_{i,p} L_r + z_r \hat{\sigma}_{i,p} L_r^{1/2}, \quad q_{i,p,t} = [S_{i,p,t} - IP_{i,p,t}]^+ \quad (6)$$

where the inventory position IP is stock on hand plus stock in transit plus outstanding orders at the DC, and z_r is a safety factor. The DC agent also uses an order-up-to policy, in one of two forms. Without shared information, it forecasts the orders it receives from stores and orders up to

$$S_{p,t}^{inst} = \hat{\mu}_p^{ord} L_m + z_d \hat{\sigma}_p^{ord} L_m^{1/2} \quad (7)$$

against its installation inventory position, which is DC stock plus production in progress plus plant backlog minus outstanding store orders. With shared information, it forecasts total point-of-sale sales, $\hat{\mu}_p$, and orders up to an echelon target.

$$S_{p,t}^{ech} = \sum_i S_{i,p,t} + \hat{\mu}_p L_m + z_d \hat{\sigma}_p L_m^{1/2} \quad (8)$$

against its echelon inventory position, which is DC stock plus production in progress plus plant backlog plus the stores' stock on hand and in transit [7], [33]. Store orders that are still outstanding are internal to the echelon and do not enter it. The plant agent produces to order from its backlog within the capacity it is allocated.

4.2. COORDINATION LAYER

The coordination layer runs twice in each day, once when the DC allocates its stock and transport and once when the plant allocates its capacity (Figure 2). It has four components.

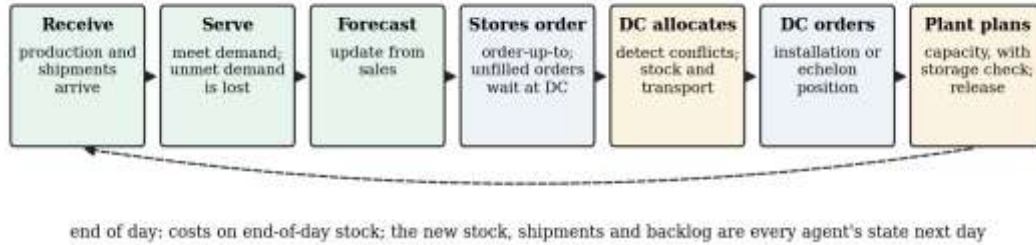


FIGURE 2 One Day of the Simulation. The Coordination Layer Acts at the Two Allocation Steps, After the Stores Have Ordered and After the DC Has Ordered. The Resulting Stock, Shipments and Backlog are Every Agent's State on the Next Day

- Shared-state aggregator: Each store posts its order together with its stock on hand and in transit, its sales forecast and its forecast error. The DC posts its stock, the space it has left and its order to the plant, and the plant posts its backlog and production in progress.
- Conflict detector: A conflict exists when orders exceed availability for a shared resource, that is, when outstanding store orders for a product exceed DC stock, when they exceed transport capacity T in total, when the plant backlog exceeds capacity K_t , or when planned production would push projected DC stock above W . When there is no conflict, every order is filled in full.
- Allocator: When a conflict exists, the allocator balances projected stock-out risk. For each store and product it computes the risk index

$$k_{i,p} = (A_{i,p} + x_{i,p} - \hat{\mu}_{i,p} L_r) / (\hat{\sigma}_{i,p} L_r^{1/2}) \quad (9)$$

where $A_{i,p}$ is the store's stock on hand and in transit. The index is the number of standard deviations of protection-period demand that the store would hold after the shipment. The allocation raises the lowest indices first, up to each store's outstanding order. It is found by water-filling. For each product, the level λ_p is the highest value at which $x_{i,p} = \min(u_{i,p}, \max(0, \hat{\sigma}_{i,p} L_r^{1/2} (\lambda_p - k_{i,p}^0)))$ satisfies constraint (1), where $k_{i,p}^0$ is the index before shipment. If the resulting total exceeds T , a common ceiling $\lambda \leq \lambda_p$ is lowered until constraint (2) holds. Under a normal approximation of protection-period demand, this equalises the stores' stock-out probabilities, which is the principle behind balanced allocation [9], [10]. Production capacity is allocated in the same way across products, using the echelon index $(E_p + y_p - \hat{\mu}_p L_e) / (\hat{\sigma}_p L_e^{1/2})$, where E_p is the physical echelon stock (DC stock, production in progress, and store stock on hand and in transit) and $L_e = L_m + L_r$.

Constraint checker Before production is released, the checker projects end-of-day DC stock on the day the production will arrive, as current end-of-day stock plus all production in progress minus expected shipments over the next L_m days. It caps total release at the space that would remain under W , and at K_t . Released quantities are removed from the backlog, oldest order first. The plan is then executed, and the resulting stock, shipments and backlog enter the next day's state.

4.3. MECHANISMS COMPARED

Table 1 lists the five mechanisms and two ablations. They share the same agents and the same store policy, and differ in the information the DC agent uses, the rule that resolves conflicts and whether the storage check is applied. The independent mechanism is the case in which each agent acts on its own information and resources are taken on a first-come, first-served basis. At the DC, stores are served in a random order each day, and each store's outstanding orders are filled in full while stock and transport last. At the plant, the oldest orders are produced first. Proportional rationing fills every outstanding order by the same fraction when a resource is short, which is common in practice and is the rule analysed by Cachon and Lariviere [12]. Information sharing replaces installation ordering with echelon ordering on shared point-of-sale data. The coordinated mechanism adds the allocator and the constraint checker to installation ordering, and the coordinated mechanism with echelon ordering adds them to information sharing. The two ablations remove the allocator or the checker from the coordinated mechanism. Both coordinated

mechanisms use shared information for allocation, namely the stores' stock positions and forecasts at the DC, and the chain's sales forecast and its error for allocating production and projecting DC stock in the storage check.

TABLE 1 Coordination Mechanisms. Each Row Keeps the Store Policy (6) and Differs in the DC Ordering Rule, The Allocation Rule at the DC and the Plant, and the Storage Check

Mechanism	DC Ordering	Allocation Of DC Stock And Transport	Allocation Of Production	Storage Check
Independent	Installation (7)	First come, first served (random store order)	Oldest order first	No
Proportional	Installation (7)	Proportional to outstanding orders	Proportional to backlog	No
Information sharing	Echelon (8)	Proportional to outstanding orders	Proportional to backlog	No
Coordinated	Installation (7)	Balanced stock-out risk (9)	Balanced echelon risk	Yes
Coordinated, echelon	Echelon (8)	Balanced stock-out risk (9)	Balanced echelon risk	Yes
Coordinated without balanced allocation	Installation (7)	Proportional to outstanding orders	Proportional to backlog	Yes
Coordinated without storage check	Installation (7)	Balanced stock-out risk (9)	Balanced echelon risk	No

5. EXPERIMENTAL DESIGN

5.1. DEMAND DATA AND PRODUCT SAMPLES

Customer demand is taken from the M5 competition data [23], which records the daily unit sales of 3,049 products in ten Walmart stores in California, Texas and Wisconsin from 29th January 2011 to 19th June 2016, including the final 28-day evaluation period. We use the 2,136 products that had recorded a sale in every one of the ten stores by 3rd October 2013, and of these, the 1,221 products that sold at least 5 units a day across the chain on average in the 90 days before 1st January 2014. For each of 20 random seeds we draw 50 of these products, so each sample has 500 store-product series. Two samples share on average 2 of their 50 products. Recorded sales are used as demand, so days on which a store was out of stock understate demand.

The simulation starts on 3rd October 2013. The first 90 days are a warm-up. The calibration year runs from 1st January to 31st December 2014 and the test period from 1st January 2015 to 19th June 2016 (536 days). Every result below is measured on the test period, and the simulation runs continuously from the start of the warm-up, so the test period begins from the state that the calibration year left behind. All samples share the same stores and the same calendar, so the confidence intervals reported below reflect variation across products, not across demand paths.

5.2. PARAMETERS AND TUNING

Table 2 lists the parameters. Costs are expressed per unit of product cost. The holding costs correspond to annual rates of about 73% at the stores and 37% at the DC, including space and handling. A lost sale costs half a unit, which gives a daily critical ratio $b/(b + h_r)$ of about 0.996, and overflow space carries a premium of three times the DC holding cost, so a unit in overflow space costs four times as much as a unit in the DC. Quantities are continuous, and the storage space counts units without regard to product volume. The shared resources are set relative to M , the mean total daily demand of the sample over the period being simulated, which is the calibration year during tuning and the test period during testing. Production capacity is $K = M/\rho_K$ and transport capacity is $T = M/\rho_T$, where ρ is the utilisation at mean demand, and storage space W is a number of days of mean demand. Setting the resources from realised demand fixes the utilisation of each scenario exactly, but it means that capacity anticipates the test period's demand level. The base case uses $\rho_K = 0.90$, $\rho_T = 0.85$ and $W = 1.5$ days, so each resource binds on some days but not on most. Across samples, M ranges from 746 to 1,441 units a day in the test period.

TABLE 2 Simulation Parameters

Parameters	Values
Stores R , products P per sample	10, 50 (500 store-product series)
Product samples (seeds)	20
Lead times L_r (DC to store), L_m (production to DC)	2 days, 4 days
Holding cost h_r (store), h_d (DC)	0.002, 0.001 per unit-day
Lost-sale cost b	0.5 per unit
Overflow premium c_o	0.003 per unit-day above W
Forecast smoothing α	0.1
Production capacity K	M/ρ_K , base $\rho_K = 0.90$
Outbound transport T	M/ρ_T , base $\rho_T = 0.85$
Storage space W	base 1.5 days of M
Warm-up, calibration, test	90 days, 2014, 1 Jan 2015 to 19 Jun 2016

The safety factors z_r and z_d change which mechanism looks best, so each mechanism is given its own. For each seed and mechanism, we choose the pair that minimises mean daily cost over the calibration year under the base resources. The search evaluates a coarse grid, $z_r \in \{1, 2, \dots, 8\}$ and $z_d \in \{-2, -1, \dots, 6\}$, and then the eight neighbours of the best point at a step of 0.5. The chosen pair is then fixed for every test scenario. Section 6.1 reports the chosen values, and Section 6.2 repeats the main comparison with equal safety factors.

5.3. SCENARIOS

The base case is followed by sweeps over one resource at a time, with production utilisation ρ_K from 0.80 to 0.98, transport utilisation ρ_T from 0.75 to 0.95 and storage space W from 1 to 3 days of demand, and by a tight case with $\rho_K = 0.95$, $\rho_T = 0.95$ and $W = 1$ day. Two events are placed at random dates in the test period, at least 60 days from either end. In the disruption scenario, the plant runs at 40% of capacity for 14 days. In the surge scenario, demand for a random 30% of products rises by 60% in the four California stores for 21 days, which the agents do not know in advance. Storage space enters decisions only through the storage check, so for mechanisms without the check the storage sweep changes only the overflow cost. To test sensitivity to the cost parameters, the base case is repeated on the first 10 samples with c_o halved and doubled and with b halved and doubled, returning the safety factors for each setting on the coarse grid.

5.4. METRICS

We report mean daily cost (5) and its components, the fill rate (units sold over units demanded), the share of product-days on which outstanding store orders exceed DC stock, the share of days on which transport or production capacity limits what is shipped or produced, and the share of days on which end-of-day DC stock exceeds W . Bullwhip is measured as the ratio of the variance of orders to the variance of demand at the same stage [5], as the median over store-product series for store orders and over products for DC orders against total store demand. We also report the variance of shipments to stores relative to their demand. Decision stability is the mean absolute day-to-day change in total DC orders divided by their mean. Communication is counted analytically as the number of values that agents and the coordination layer exchange per day, and computation as the time spent per simulated day in the DC's allocation of stock and transport and in the plant's allocation and release of production. Differences between mechanisms are reported as the mean of the per-sample percentage differences, with a 95% confidence interval and a one-sample t-test on those differences over the 20 samples. For the main comparison across the 14 scenarios, we also apply a Holm correction.

6. RESULTS

6.1. BASE CASE

Table 3 reports the base case over 20 product samples. Mean total demand was about 1,030 units a day, so a daily cost of 21 corresponds to about 2% of the product cost of the units sold. How conflicts were resolved when agents acted on their own mattered most. Proportional rationing cost 18.6% less than independent agents (95% CI 17.0% to 20.2%, lower in all 20 samples). Most of the difference was lost sales (9.40 against 6.03) and overflow (1.54 against 0.27). Serving stores in a random order each day filled some stores completely while others received nothing, and the DC, forecasting the resulting uneven orders, held 1.9 days of stock against 1.1 days.

TABLE 3 Base Case Over 20 Product Samples ($P_k = 0.90$, $P_t = 0.85$, $W = 1.5$ Days). Daily Cost is the Mean Daily Cost (5) With Its 95% Confidence Interval across Samples, And its Components are Holding, Lost Sales and Overflow. DC and Store Stock Are Mean End-Of-Day Stock in Days Of Total Demand. Z_r and Z_d are the Median Safety Factors Chosen on the Calibration Year

Mechanism	Daily cost	Holding	Lost sales	Overflow	Fill rate (%)	DC stock (days)	Store stock (days)	Z_r, Z_d
Independent	26.18 $\pm$ 2.60	15.23	9.40	1.54	98.20	1.89	6.50	4.5, 1.5
Proportional	21.30 $\pm$ 2.25	15.00	6.03	0.27	98.84	1.08	6.76	5, 0.5
Information sharing	23.11 $\pm$ 2.41	14.57	7.79	0.75	98.50	1.42	6.40	5, 2.25
Coordinated	21.37 $\pm$ 2.01	15.10	6.13	0.14	98.82	1.00	6.85	5.5, 0.5
Coordinated, echelon	22.95 $\pm$ 2.18	14.98	7.76	0.20	98.50	1.14	6.73	5.25, 2
Coordinated without balanced allocation	21.45 $\pm$ 2.38	15.18	6.12	0.15	98.83	1.02	6.87	5, 1
Coordinated without storage check	21.01 $\pm$ 1.89	14.90	5.85	0.27	98.87	1.12	6.70	5, 0.5

Information sharing with echelon ordering cost 8.6% more than proportional rationing with installation ordering (95% CI 6.3% to 10.8%, higher in all 20 samples). It did reduce the variability of the DC's orders (Table 6), but it held more stock at the DC (1.42 against 1.08 days), less at the stores (6.40 against 6.76 days), and lost 29% more sales. We tested two explanations with separate runs on all 20 samples, retuning both mechanisms on the coarse grid in each case (with this tuning, the gap in the base case was 9.1%). The first is the balance assumption behind echelon policies [7], [10]. The echelon position counts all stock at the stores, including stock of slow sellers that one store holds in excess and that cannot be moved to another store that is short, so the DC orders too little of those products. When the ten stores were merged into a single store, so that no stock could be stranded, echelon ordering cost 4.4% less than installation ordering (95% CI 3.0% to 5.8%, lower in 18 of 20 samples). The second explanation is censoring, because agents forecast from sales, which fall when stores are out of stock. When all agents forecast from true demand, echelon ordering still costs 5.7% more (95% CI 3.6% to 7.8%, higher in all 20 samples). Stock stranded at the wrong store is therefore likely the main reason, and censored sales account for somewhat over a third of the gap. Merging the stores also pools their demand, so the first test does not isolate imbalance completely.

The coordinated mechanism, which adds balanced allocation and the storage check to installation ordering, costs the same as proportional rationing in the base case (difference +0.7%, 95% CI -1.4% to 2.8%, $p = 0.47$). The coordinated mechanism with echelon ordering cost the same as information sharing (-0.4%, $p = 0.67$). The tuned safety factors were high at the stores, with a median z_t of 4.5 to 5.5, because a lost sale costs 250 times as much as a day of store holding. All chosen values were inside the search grid, and the cost at the nearest coarse-grid points was a median of 0.4% to 2.0% above the optimum, depending on the mechanism.

6.2. WHEN COORDINATION MATTERS

Table 4 reports cost in every scenario, and Figure 3 shows the three resource sweeps. Independent agents cost the most in every scenario, and information sharing costs more than proportional rationing in every scenario. The coordinated mechanism's advantage over proportional rationing depended on whether production or transport capacity was close to its limit. With production at 95% utilisation it cost 4.7% less ($p < 0.001$, lower in 17 of 20 samples); at 98% it cost 7.8% less (19 of 20); and with transport at 95% it cost 11.9% less (95% CI 9.7% to 14.1%, 19 of 20). These three differences remain significant after a Holm correction over the 14 scenarios. In the tight case, the difference was 4.1% ($p = 0.030$, not significant after correction), and with transport at 90% it was 1.6% ($p = 0.096$). With slack production and transport, and in the surge scenario, the two mechanisms cost the same. With one day of storage space, the coordinated mechanism cost 5.9% more ($p = 0.006$, 0.068 after correction), for a reason that Section 6.4 traces to the storage check.

TABLE 4 Mean Daily Cost by Scenario (20 Product Samples) for the Independent (Indep.), Proportional (Prop.), Information Sharing (Info), Coordinated (Coord.) and Coordinated with Echelon Ordering (Coord. E) Mechanisms. The Last Three Columns Compare the Coordinated Mechanism with Proportional Rationing, With the Mean Paired Difference in Percent and its 95% Confidence Interval, The P-Value of a T-Test on the Paired Differences and its Holm-Adjusted Value Over the 14 Scenarios, And the Number of Samples in Which the Coordinated Mechanism Cost Less

Scenario	Indep.	Prop.	Info	Coord.	Coord. E	Coord. vs Prop. (%)	p	Holm p	Lower
Base (ρ_K 0.90, ρ_T 0.85, W 1.5)	26.18	21.30	23.11	21.37	22.95	+0.7 $\pm$ 2.1	0.469	1.000	8/20
$\rho_K = 0.80$	25.25	20.57	22.83	20.58	22.37	+0.2 $\pm$ 1.6	0.828	1.000	13/20
$\rho_K = 0.85$	25.20	20.44	22.57	20.69	22.42	+1.3 $\pm$ 1.5	0.102	0.863	6/20
$\rho_K = 0.95$	29.98	25.24	26.30	23.90	25.12	-4.7 $\pm$ 2.0	< 0.001	< 0.001	17/20
$\rho_K = 0.98$	33.96	29.54	30.31	27.05	28.18	-7.8 $\pm$ 2.2	< 0.001	< 0.001	19/20
$\rho_T = 0.75$	25.76	21.27	23.02	21.43	22.99	+1.1 $\pm$ 2.2	0.310	1.000	8/20
$\rho_T = 0.90$	27.65	21.73	23.73	21.27	22.85	-1.6 $\pm$ 2.0	0.096	0.863	12/20
$\rho_T = 0.95$	33.53	26.82	28.91	23.44	24.87	-11.9 $\pm$ 2.2	< 0.001	< 0.001	19/20
$W = 1.0$ day	27.44	21.86	23.98	23.16	24.88	+5.9 $\pm$ 4.0	0.006	0.068	5/20
$W = 2.0$ days	25.32	21.10	22.64	20.99	22.46	0.0 $\pm$ 1.9	0.979	1.000	8/20
$W = 3.0$ days	24.70	21.03	22.38	20.91	22.35	-0.1 $\pm$ 1.8	0.901	1.000	8/20
Tight (ρ_K 0.95, ρ_T 0.95, W 1.0)	34.24	26.79	28.82	25.57	27.20	-4.1 $\pm$ 3.7	0.030	0.305	16/20
Production disruption	30.40	24.97	26.50	24.65	26.22	-0.9 $\pm$ 2.5	0.462	1.000	14/20
Demand surge	26.80	21.80	23.55	21.81	23.40	+0.5 $\pm$ 2.1	0.654	1.000	10/20

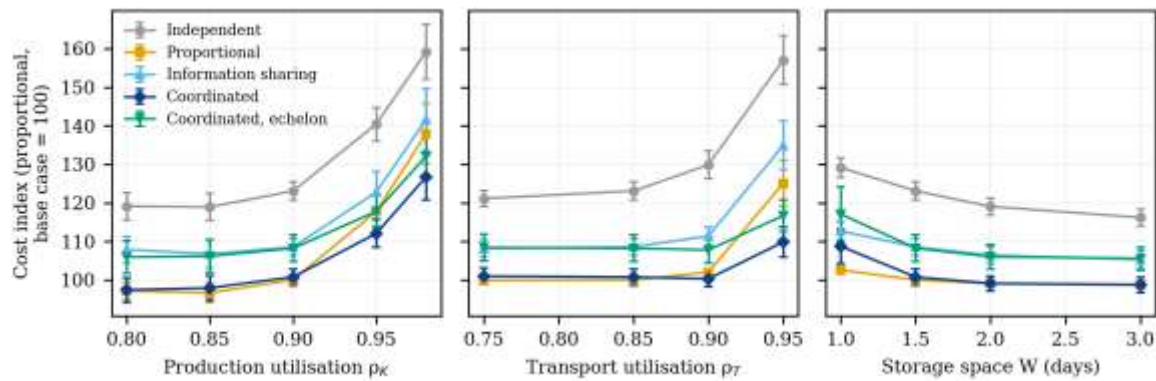


FIGURE 3 Cost of the Five Mechanisms as One Shared Resource is Tightened at a Time, With the Other Two at their Base Values. Cost is Indexed to Each Sample's Proportional-Rationing Cost in the Base Case (= 100). Error Bars are 95% Confidence Intervals across the 20 Samples. Storage Space Affects Decisions Only through the Storage Check, So for the Mechanisms without it the Right Panel Reflects Overflow Cost Alone

A similar pattern held between the two mechanisms with echelon ordering. The coordinated mechanism with echelon ordering cost 3.8%, 6.3% and 13.2% less than information sharing at 95% and 98% production utilisation and 95% transport utilisation (all $p \leq 0.004$). It also cost 1.9% less at 80% production utilisation ($p = 0.016$, 0.16 after correction), a smaller advantage with slack capacity that does not fit the pattern. Because each mechanism has its own tuned safety factors, we repeated the comparison with the coordinated mechanism run at proportional rationing's safety factors. The differences were almost unchanged, at -4.6%, -7.4% and -11.6% in the same three scenarios and +0.2% in the base case, so in these scenarios the advantage comes from the allocation rule and not from different safety stocks. In the tight case, the difference at equal safety factors was smaller (-2.6%, $p = 0.12$), and for the echelon pair at 95% production utilisation it fell to -2.4% ($p = 0.06$).

6.3. DISRUPTION AND DEMAND SURGE

In the disruption scenario, the plant ran at 40% of capacity for 14 days. Figure 4 shows the daily fill rate around the event, averaged over the 20 samples. Fill rates began to fall about a week into the disruption, when stock in the pipeline had been used, and reached their lowest point about 19 days after it began, once the last reduced production had reached the stores. During the 14 days of reduced production, the coordinated mechanism kept a fill rate of 95.4%, against 93.7% for proportional rationing ($p = 0.005$, higher in 15 of 20 samples), 94.2% for information sharing and 92.2% for independent agents. Balanced allocation gave scarce production first to the products whose echelon stock was lowest, which delayed the fall in fill rate. Recovery over the following 28 days was similar for all mechanisms except independent agents (88% to 89% against 85%). Over the whole test period, however, the coordinated mechanism's cost was not significantly different from proportional rationing (-0.9%, $p = 0.46$), because the storage check offset part of the gain (Section 6.4).

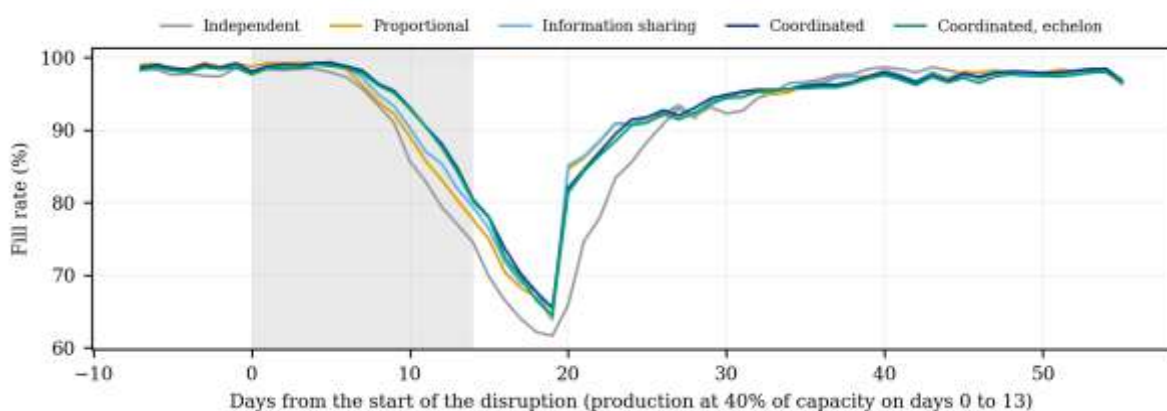


FIGURE 4 Daily Fill Rate Around a 14-Day Production Disruption (Shaded), Averaged Over the 20 Product Samples. Day 0 is the First Day of Reduced Production

The regional demand surge raised demand for 30% of products by 60% in the four California stores for 21 days. It created competition between stores for DC stock, but all mechanisms except independent agents absorbed it. During the surge, the fill rate was 97.8% for the coordinated mechanism and 97.5% for proportional rationing ($p = 0.40$), against 96.2% for independent agents.

6.4. ABLATION

Table 5 changes one component of the coordinated mechanism at a time. Balanced allocation was the component that mattered. Without it, cost rose by 5.9% and 9.6% with production at 95% and 98% utilisation, by 11.6% with transport at 95% and by 9.3% in the tight case (all $p < 0.001$), and it made no difference with slack resources. Used alone, without the storage check, balanced allocation cost 5.8%, 8.6%, 7.8% and 8.9% less than proportional rationing in the same four scenarios, 2.9% less in the disruption and 3.0% less with transport at 90% (all $p \leq 0.003$), and the same in the base case.

TABLE 5 Ablation of the Coordinated Mechanism (20 Product Samples). Each Column Changes one Component and Reports the Paired Change in Mean Daily Cost Relative to the Full Coordinated Mechanism, With its 95% Confidence Interval and P-Value. In the First two Columns, Positive Values Mean that the Component Lowers Cost. In the Last Column, Positive Values Mean that Echelon Ordering Raises Cost

Scenario	Coordinated cost	Without balanced allocation (%)	Without storage check (%)	With echelon ordering (%)
Base (ρ_K 0.90, ρ_T 0.85, W 1.5)	21.37	0.0 ± 2.3 ($p = 0.967$)	-1.5 ± 1.0 ($p = 0.006$)	$+7.4 \pm 2.2$ ($p < 0.001$)
$\rho_K = 0.95$	23.90	$+5.9 \pm 2.4$ ($p < 0.001$)	-1.2 ± 0.9 ($p = 0.016$)	$+5.2 \pm 2.3$ ($p < 0.001$)
$\rho_K = 0.98$	27.05	$+9.6 \pm 2.9$ ($p < 0.001$)	-0.8 ± 0.9 ($p = 0.061$)	$+4.4 \pm 2.9$ ($p = 0.005$)
$\rho_T = 0.95$	23.44	$+11.6 \pm 2.5$ ($p < 0.001$)	$+4.8 \pm 1.8$ ($p < 0.001$)	$+6.3 \pm 2.2$ ($p < 0.001$)
$W = 1.0$ day	23.16	-0.7 ± 4.5 ($p = 0.743$)	-5.8 ± 3.1 ($p < 0.001$)	$+7.6 \pm 4.1$ ($p < 0.001$)
Tight (ρ_K 0.95, ρ_T 0.95, W 1.0)	25.57	$+9.3 \pm 4.5$ ($p < 0.001$)	-4.6 ± 2.9 ($p = 0.003$)	$+6.7 \pm 4.1$ ($p = 0.003$)
Production disruption	24.65	$+2.9 \pm 2.1$ ($p = 0.011$)	-1.9 ± 1.4 ($p = 0.014$)	$+6.6 \pm 2.9$ ($p < 0.001$)

The storage check lowered cost only when transport was the bottleneck. With transport at 95% utilisation, shipments were limited by the dock rather than by stock, stock built up at the DC, and removing the check raised cost by 4.8% ($p < 0.001$). In most other scenarios removing the check lowered cost slightly, and with one day of storage space or in the tight case removing it lowered cost by 5.8% and 4.6%. The check holds back production whenever projected DC stock would exceed W . In the one-day case it did so on 44% of days, and the production it deferred was later missing at the stores. Overflow cost fell from 0.85 to 0.21, but lost-sale cost rose from 5.85 to 8.81. A hard cap on production is too blunt when overflow space is available at a moderate premium. Replacing installation ordering with echelon ordering raised cost by 4.4% to 7.6% in every scenario in Table 5.

6.5. STABILITY, COMMUNICATION, COMPUTATION AND SCALE

Table 6 reports how the mechanisms behaved in the base case. Store orders varied about 2.2 to 2.3 times as much as store demand under every mechanism, because all stores used the same policy. DC orders varied 4.4 to 4.8 times as much as total demand with installation ordering and 4.0 to 4.1 times with echelon ordering, so shared information reduced the bullwhip effect at the DC [3], [5], as expected, even though it raised cost. Balanced allocation made shipments to stores more variable (1.9 against 1.6 times demand), because it sends stock to whichever stores are lowest on a given day. Conflicts were frequent. Outstanding store orders exceeded DC stock on 38% of product-days with proportional rationing and on 51% with the coordinated mechanism, whose storage check kept the DC leaner, and the plant backlog exceeded capacity on 63% and 72% of days.

In Table 6 Behaviour of the mechanisms in the base case (means over 20 samples). Variance ratios are medians over store-product series (store orders and shipments against store demand) or over products (DC orders against total store demand). DC order change is the mean absolute day-to-day change in total DC orders as a share of their mean. Stock conflicts are product-days on which outstanding store orders exceeded DC stock. Transport binding counts days on which transport limited shipments, and storage check binding counts days on which the check reduced production. Values exchanged per day are counted analytically with $R = 10$ and $P = 50$, as $2RP + 2P$ for the first two mechanisms (store orders and shipments, DC orders and releases), $5RP + 2P$ for information sharing (adding each store's stock position, target and sales), and $6RP + 4P + 1$ for both coordinated mechanisms (adding each store's stock position, forecast, forecast error and sales, from which the layer also keeps a chain-level sales forecast, together with DC stock and space and plant backlog). Allocation time covers the DC's allocation of stock and transport and the plant's allocation and release of production, in single-threaded Python.

TABLE 6 Comparative Performance Measures Across Different Inventory Coordination Strategies

Measures	Indep.	Prop.	Info	Coord.	Coord. E
Store orders / demand variance	2.19	2.30	2.20	2.34	2.32
Store shipments / demand variance	2.35	1.61	1.61	1.95	1.90

DC orders / demand variance	4.81	4.40	3.98	4.36	4.09
DC order change (% of mean)	31.7	30.0	28.5	30.2	29.0
Stock conflicts (% of product-days)	18.9	37.5	33.3	51.0	53.9
Transport binding (% of days)	27.1	17.0	22.5	22.4	22.5
Backlog above capacity (% of days)	62.0	63.0	61.0	72.1	79.3
Storage check binding (% of days)	0.0	0.0	0.0	18.6	30.7
DC stock above W (% of days)	69.8	21.6	41.6	15.6	23.6
Values exchanged per day	1,100	1,100	2,600	3,201	3,201
Allocation time (ms per day)	0.19	0.17	0.18	0.99	1.01

The coordinated mechanism exchanged 3,201 values per day against 1,100 for proportional rationing, and its allocation steps took about 1.0 ms per simulated day against 0.2 ms. To check how this scales, we reran the base case on five new product samples with 25 to 400 products and 10 stores (250 to 4,000 store-product series), and with 50 products and 2 or 5 stores, using the median safety factors of Table 3 without retuning. Allocation time for the coordinated mechanism grew from 0.9 ms to 1.7 ms per day across these sizes, because the water-filling steps are vectorised and use a fixed number of bisection steps. Some relations held at every size. Proportional rationing cost 17% to 22% less than independent agents on average, and information sharing cost 8% to 15% more than proportional rationing. The coordinated mechanism cost between 1.9% less and 6.5% more than proportional rationing on average, and it was the cheapest mechanism only with five stores. With 100 to 400 products, it cost 1.6% to 3.1% more than proportional rationing, and with 400 products it cost more in all five samples, which is consistent with the base-case cost of the storage check (Section 6.4). The order of the mechanisms with echelon ordering varied with size. With 25 products, where per-sample differences between the coordinated mechanism and proportional rationing ranged from -13% to +41% and the untuned safety factors fit least well, the coordinated mechanism with echelon ordering cost slightly more than independent agents.

6.6. SENSITIVITY TO COSTS

Table 7 repeats the base case on the first 10 samples with the overflow premium and the lost-sale cost halved and doubled, retuning the safety factors for each setting. Information sharing cost more than proportional rationing in every setting, by 5.4% to 17.7% (all $p \leq 0.01$). The coordinated mechanism was not cheaper than proportional rationing in any cost setting of the base case, and with a doubled lost-sale cost it was 5.9% more expensive ($p = 0.028$). This is consistent with Section 6.4. In the base case balanced allocation has little to do, and the storage check trades overflow for lost sales, which is a poor trade when lost sales are expensive.

TABLE 7 Sensitivity of the Base Case to the Cost Parameters (First 10 Product Samples, Safety Factors Retuned for Each Setting on the Coarse Grid). The First Row gives the Base Costs on the Same 10 Samples With the Same Coarse Tuning

Cost setting	Proportional cost	Coordinated vs Proportional (%)	P	Information sharing vs Proportional (%)	P
Base costs	19.61	+1.6 ± 1.8	0.082	+10.3 ± 4.4	< 0.001
$c_o = 0.0015$ (halved)	19.45	+1.7 ± 1.8	0.053	+11.7 ± 8.1	0.010
$c_o = 0.006$ (doubled)	20.02	0.0 ± 2.1	0.998	+9.0 ± 4.4	0.001
$b = 0.25$ (halved)	16.13	+0.4 ± 1.1	0.435	+5.4 ± 2.4	< 0.001
$b = 1.0$ (doubled)	23.72	+5.9 ± 5.1	0.028	+17.7 ± 11.0	0.006

7. DISCUSSION

The results give an order of priorities for coordinating agents that share inventory and capacity, and they run against some common expectations. The first priority is the rule for rationing scarce stock and capacity. Serving whoever comes first cost about 23% more than proportional rationing, because it concentrated shortages on some stores and made the DC's demand signal more erratic. Any coordination layer should at least replace first come, first served with a rule that spreads shortages.

The second finding is that sharing point-of-sale data for echelon ordering raised cost in this setting, even though it damped the DC's bullwhip. This agrees with earlier findings that the value of shared information is often small and depends on the setting [6], [32], and with the known weakness of echelon policies when stock downstream cannot be rebalanced [10]. In a chain whose stores sell many slow products and lose unmet demand, the echelon position overstates how much stock is usable, and when the stores were merged so that stock could not be stranded, echelon ordering became the cheaper rule. Installation ordering on real store orders was the better base, and in this setting shared information was more useful for allocation than for ordering.

The third finding concerns the coordination layer itself. Conflict-aware allocation, which gives scarce stock, transport and production to the stores and products at greatest risk of stock-out, paid off when production ran at 95% to 98% of capacity or transport at 90% to 95%, where on its own it cut cost by 3% to 9% relative to proportional rationing, and during a disruption, where the coordinated mechanism kept the fill rate 1.7 points higher. With spare capacity, it made no difference, because the

conflicts it resolves were small. The storage check, by contrast, helped only when transport was the bottleneck and hurt when storage was tight, because it deferred production that the stores later needed. As a result, balanced allocation without the storage check was the cheapest configuration in 13 of the 14 scenarios, all except the one in which transport was the bottleneck, and it was never significantly worse than proportional rationing. Explicit constraint checks are useful, but a check that enforces a soft limit as a hard cap can cost more than the violation it prevents. A practical reading is to use balanced allocation wherever a resource is regularly near its limit, and to apply storage checks only with a rule that weighs the cost of overflow against the cost of deferred production.

These findings concern how a coordination layer resolves the orders of agents that follow fixed order-up-to policies. The same layer could sit behind learning or language-model agents, whose orders are harder to predict [18], [22], but whether its allocation and checks help such agents in the same way has not been tested here.

The communication cost of coordination is modest. The coordinated mechanism exchanges about three times as many values per day as proportional rationing, because each store posts its stock position, forecast, forecast error and sales with its order, and it needs about a millisecond of computation per day for 500 store-product series. These costs assume that stores are willing to share their state with the coordination layer.

8. LIMITATIONS AND FUTURE WORK

The study has several limitations. The network has one plant, one DC and ten stores. Real networks have several DCs, transshipment between stores and several plants and transshipment in particular would ease the imbalance that hurt echelon ordering here. The test in which the stores were merged also pools their demand, so it does not isolate imbalance completely. Demand is recorded sales, which understate demand on days when a store was out of stock, and it is treated as independent of the decisions studied here. All samples share the same stores and calendar, so the confidence intervals reflect variation across products and not across demand paths. All agents follow order-up-to policies with exponentially smoothed forecasts and normal approximations, which fit slow-moving store-level series only roughly. Quantities are continuous, storage space counts units without regard to volume, lead times are fixed, and capacity has no setups or batch sizes. Agents that learn their policies, negotiate or act strategically, for example by inflating orders when they expect to be rationed [11], [12], may respond differently to each mechanism, and our proportional mechanism assumes truthful orders, which favours it. The shared resources were scaled to the realised demand of the period being simulated, so capacity anticipated the level of test-period demand. Safety factors were tuned once, under the base resources and costs, and kept fixed across scenarios, which is how an operator facing an unexpected disruption would be placed, but retuning per scenario could change some differences. The storage check uses a simple projection and a hard cap, and Section 6.4 shows that this is its main weakness. Finally, we do not compare against a centralised planner that optimises over the whole horizon, so the distance of each mechanism from the best achievable cost is unknown.

Future work should replace the hard storage cap with a check that weighs overflow cost against deferred production, extend the network to several DCs with transshipment, and add a rolling-horizon optimisation benchmark. It should also place learning or language-model agents behind the same coordination layer and test whether its allocation and checks keep their plans feasible and stable.

9. CONCLUSION

This paper studied how software agents that share DC stock, outbound transport, DC storage and production capacity should be coordinated. It formulated the shared-resource constraints and proposed a coordination layer that aggregates agent state, detects conflicts, allocates scarce resources to balance projected stock-out risk, and checks planned production against storage space. In a simulation driven by the real daily sales of 50 products in ten Walmart stores, over 20 product samples and 536 test days, three results stand out. First, the rationing rule matters most when agents act independently, and proportional rationing cost 18.6% less than first-come, first-served. Second, echelon ordering on shared point-of-sale data reduced the DC's order variability but raised cost by 8.6%, most likely because stock held at one store could not serve another, since it became the cheaper rule when the stores were merged. Third, the full layer added nothing when capacity was slack, but cut cost by 4.7% to 11.9% when production or transport ran at 95% utilisation or more and kept service higher during a production disruption. Balanced allocation produced most of this gain, while the hard storage check helped only when transport was the bottleneck and hurt when storage was tight. Balanced allocation without the storage check was the cheapest configuration in every scenario except the one in which transport was the bottleneck. For practitioners, the results suggest replacing first come, first served with at least proportional rationing, checking whether downstream stock can be rebalanced before relying on echelon ordering, using balanced allocation wherever a resource regularly runs near its limit, and designing constraint checks that weigh the cost of a violation against the cost of preventing it.

DECLARATIONS

- Funding. This research received no external funding.
- Competing Interests. The author declares no competing interests.
- Ethics Approval. Not applicable. The study uses publicly available sales data and involves no human participants.

- Data Availability. The M5 competition data are publicly available [23]. The simulation code, product samples and results are available from the author on request.
- Use Of Generative AI. Generative AI tools were used to assist with coding the experiments and editing the text. The author reviewed all content and takes full responsibility for it.

REFERENCES

- [1] O. Kwon, G. P. Im and K. C. Lee, "MACE-SCM: A multi-agent and case-based reasoning collaboration mechanism for supply chain management under supply and demand uncertainties," *Expert Systems with Applications*, vol. 33, no. 3, pp. 690–705, Oct. 2007, doi: 10.1016/j.eswa.2006.06.015
- [2] A. Surana, S. Kumara, M. Greaves, and U. N. Raghavan, "Supply-chain networks: a complex adaptive systems perspective," *International Journal of Production Research*, vol. 43, no. 20, pp. 4235–4265, Oct. 2005, doi: 10.1080/00207540500142274
- [3] H. L. Lee, V. Padmanabhan, and S. Whang, "Information Distortion in a Supply Chain: The Bullwhip Effect," *Management Science*, vol. 50, no. 12_supplement, pp. 1875–1886, Dec. 2004, doi: 10.1287/mnsc.1040.0266
- [4] J. D. Sterman, "Modeling Managerial Behavior: Misperceptions of Feedback in a Dynamic Decision Making Experiment," *Management Science*, vol. 35, no. 3, pp. 321–339, Mar. 1989, doi: 10.1287/mnsc.35.3.321
- [5] F. Chen, Z. Drezner, J. K. Ryan, and D. Simchi-Levi, "Quantifying the Bullwhip Effect in a Simple Supply Chain: The Impact of Forecasting, Lead Times, and Information," *Management Science*, vol. 46, no. 3, pp. 436–443, Mar. 2000, doi: 10.1287/mnsc.46.3.436.12069
- [6] G. P. Cachon and M. Fisher, "Supply Chain Inventory Management and the Value of Shared Information," *Management Science*, vol. 46, no. 8, pp. 1032–1048, Aug. 2000, doi: 10.1287/mnsc.46.8.1032.12029
- [7] A. J. Clark and H. Scarf, "Optimal Policies for a Multi-Echelon Inventory Problem," *Management Science*, vol. 6, no. 4, pp. 475–490, July 1960, doi: 10.1287/mnsc.6.4.475
- [8] Sven Axäter, *Inventory Control*. Springer International Pu, 2015.
- [9] Gary Eppen and Linus Schrage, "Centralized ordering policies in a multi-warehouse system with lead times and random demand," *Multi-Level Production/Inventory Control Systems: Theory and Practice*, L. B. Schwarz, Ed., TIMS Studies in the Management Sciences, vol. 16. Amsterdam, The Netherlands: North-Holland, pp. 51–67, 1981.
- [10] M. C. van der Heijden, E. B. Diks and A. G. de Kok, "Stock allocation in general multi-echelon distribution systems with (R, S) order-up-to-policies," *International Journal of Production Economics*, vol. 49, no. 2, pp. 157–174, Apr. 1997, doi: 10.1016/s0925-5273(97)00005-4.
- [11] G. P. Cachon and M. A. Lariviere, "Capacity Choice and Allocation: Strategic Behavior and Supply Chain Performance," *Management Science*, vol. 45, no. 8, pp. 1091–1108, Aug. 1999, doi: 10.1287/mnsc.45.8.1091.
- [12] G. P. Cachon and M. A. Lariviere, "An equilibrium analysis of linear, proportional and uniform allocation of scarce capacity," *IIE Transactions*, vol. 31, no. 9, pp. 835–849, Sept. 1999, doi: 10.1080/07408179908969885
- [13] J. M. Swaminathan, S. F. Smith, and N. M. Sadeh, "Modeling Supply Chain Dynamics: A Multiagent Approach," *Decision Sciences*, vol. 29, no. 3, pp. 607–632, July 1998, doi: 10.1111/j.1540-5915.1998.tb01356.x
- [14] M. S. Fox, M. Barbuceanu and R. Teigen, "Agent-oriented supply-chain management," *International Journal of Flexible Manufacturing Systems*, vol. 12, no. 2/3, pp. 165–188, 2000, doi: 10.1023/a:1008195614074
- [15] R. G. Smith, "The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver," *IEEE Transactions on Computers*, vol. C-29, no. 12, pp. 1104–1113, Dec. 1980, doi: 10.1109/tc.1980.1675516
- [16] X. Liming, S. Mak, M. Minaricova, and A. Brintrup, "On implementing autonomous supply chains: A multi-agent system approach," *Computers in Industry*, vol. 161, p. 104120, Oct. 2024, doi: 10.1016/j.compind.2024.104120.
- [17] B. Rolf, I. Jackson, M. Müller, S. Lang, T. Reggelin, and D. Ivanov, "A review on reinforcement learning algorithms and applications in supply chain management," *International Journal of Production Research*, vol. 61, no. 20, pp. 1–29, Nov. 2022, doi: 10.1080/00207543.2022.2140221
- [18] A. Oroojlooyjadid, M. Nazari, L. V. Snyder, and M. Takáč, "A Deep Q-Network for the Beer Game: Deep Reinforcement Learning for Inventory Optimization," *Manufacturing & Service Operations Management*, Feb. 2021, doi: 10.1287/msom.2020.0939.
- [19] Y. Yang, M. Wang, J. Wang, P. Li, and M. Zhou, "Multi-Agent Deep Reinforcement Learning for Integrated Demand Forecasting and Inventory Optimization in Sensor-Enabled Retail Supply Chains," *Sensors*, vol. 25, no. 8, p. 2428, Apr. 2025, doi: 10.3390/s25082428.
- [20] B. Li, Konstantina Mellou, B. Zhang, Jeevan Pathuri, and Ishai Menache, "Large Language Models for Supply Chain Optimization," *arXiv (Cornell University)*, July 2023, doi: 10.48550/arxiv.2307.03875
- [21] Y. Quan and Z. Liu, "InvAgent: A Large Language Model based Multi-Agent System for Inventory Management in Supply Chains," *arXiv (Cornell University)*, July 2024, doi: 10.48550/arxiv.2407.11384.
- [22] X. Zhao, Y. Xie, C. Chen and Y. Sun, "AIM-Bench: Evaluating decision-making biases of agentic LLM as inventory manager," *arXiv:2508.11416*, 2025.
- [23] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "M5 accuracy competition: Results, findings, and conclusions," *International Journal of Forecasting*, vol. 38, no. 4, Jan. 2022, doi: 10.1016/j.ijforecast.2021.11.013.
- [24] K. Lee, W. Kim, and M. Kim, "Supply Chain Management using Multi-agent System," *Lecture Notes in Computer Science*, pp. 215–225, 2004, doi: 10.1007/978-3-540-30104-2_16
- [25] Y. Fu, Rajesh Piplani, D. de, and J.-R. Wu, "Multi-agent enabled modeling and simulation towards collaborative inventory management in supply chains," *Winter Simulation Conference*, Dec. 2000, doi: 10.1109/wsc.2000.899167
- [26] J. Du, V. Sugumaran, and B. Gao, "RFID and Multi-Agent Based Architecture for Information Sharing in Prefabricated Component Supply Chain," *IEEE Access*, vol. 5, pp. 4132–4139, 2017, doi: 10.1109/access.2017.2665778
- [27] M. Giannakis and M. Louis, "A multi-agent based system with big data processing for enhanced supply chain agility," *Journal of Enterprise Information Management*, vol. 29, no. 5, pp. 706–727, Sept. 2016, doi: 10.1108/jeim-06-2015-0050
- [28] David Pardoe and Peter Stone, "An autonomous agent for supply chain management," in *Business Computing*, G. Adomavicius and A. Gupta, Eds., *Handbooks in Information Systems*, Emerald Group Publishing, vol. 3, pp. 141–172, 2009.

- [29] M. Herrera, M. Pérez-Hernández, A. Kumar Parlikad, and J. Izquierdo, "Multi-Agent Systems and Complex Networks: Review and Applications in Systems Engineering," *Processes*, vol. 8, no. 3, p. 312, Mar. 2020, doi: 10.3390/pr8030312.
- [30] X. Li and Q. Wang, "Coordination mechanisms of supply chain systems," *European Journal of Operational Research*, vol. 179, no. 1, pp. 1–16, May 2007, doi: 10.1016/j.ejor.2006.06.023
- [31] J. C. Hennet and Y. Arda, "Supply chain coordination: A game-theory approach," *Engineering Applications of Artificial Intelligence*, vol. 21, no. 3, pp. 399–405, Apr. 2008, doi: 10.1016/j.engappai.2007.10.003.
- [32] S. Gavirneni, R. Kapuscinski, and S. Tayur, "Value of Information in Capacitated Supply Chains," *Management Science*, vol. 45, no. 1, pp. 16–24, Jan. 1999, doi: 10.1287/mnsc.45.1.16
- [33] E. A. Silver, D. F. Pyke and D. J. Thomas, *Inventory and Production Management in Supply Chains*, 4th ed. Boca Raton, FL, USA: CRC Press, 2017.
- [34] Ö. Özer, "Inventory management: Information, coordination, and rationality," in *Planning Production and Inventories in the Extended Enterprise: A State of the Art Handbook*, vol. 1, K. G. Kempf, P. Keskinocak and R. Uzsoy, Eds. New York, NY, USA: Springer, pp. 321–365, 2011.
- [35] M. C. Gurbuz, *Coordinated Replenishment Strategies in Multi-item Inventory/distribution Systems*, 2007.
- [36] G. Flidner, "CPFR: an emerging supply chain tool," *Industrial Management & Data Systems*, vol. 103, no. 1, pp. 14–21, Feb. 2003, doi: 10.1108/02635570310456850.
- [37] J. A. Muckstadt, D. H. Murray, J. A. Rappold, and D. E. Collins, "Guidelines for Collaborative Supply Chain System Design and Operation," *Information Systems Frontiers*, vol. 3, no. 4, pp. 427–453, 2001, doi: 10.1023/a:1012824820895
- [38] S. C. Graves and S. P. Willems, "Optimizing Strategic Safety Stock Placement in Supply Chains," *Manufacturing & Service Operations Management*, vol. 2, no. 1, pp. 68–83, Jan. 2000, doi: 10.1287/msom.2.1.68.23267
- [39] L. S. Dias and M. G. Ierapetritou, "From process control to supply chain management: An overview of integrated decision making strategies," *Computers & Chemical Engineering*, vol. 106, pp. 826–835, Nov. 2017, doi: 10.1016/j.compchemeng.2017.02.006.
- [40] T. Rashid, Mikayel Samvelyan, C. Schroeder, G. Farquhar, J. Foerster, and Shimon Whiteson, "QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning," Mar. 2018, doi: 10.48550/arxiv.1803.11485.
- [41] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 6382 - 6393, 2017.
- [42] J. Foerster, G. Farquhar, T. Afouras, N. Nardelli and S. Whiteson, "Counterfactual multi-agent policy gradients," *AAAI'18/IAAI'18/EAAI'18: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, vol. 32, no. 1, pp. 2974 – 2982, Feb 2018.
- [43] Z. Ji et al., "Survey of Hallucination in Natural Language Generation," *ACM Computing Surveys*, vol. 55, no. 12, Nov. 2022, doi: 10.1145/3571730.