Attributing Inventory Performance via Shapley-Based Counterfactual Decomposition

Lu Xu *

Posted Date: 19 May 2026

doi: 10.20944/preprints202603.2499.v2

Keywords: inventory management; Shapley value; counterfactual decomposition; performance attribution

 doi:10.20944/preprints202603.2499.v2

Article

Attributing Inventory Performance via Shapley-Based Counterfactual Decomposition

Lu Xu

Independent Researcher, USA; shufesifexulu@gmail.com

Abstract

Inventory systems are typically evaluated using aggregate performance metrics such as out-of-stock and average inventory. In supply chain management, it is important to understand the underlying reasons for a period's performance— specifically, how previous inventory management decisions, such as order placement, lead to the result and what their contributions are. Traditional methods are often restrictive and cannot be applied to broader cases. This paper proposes a Shapley-based decomposition framework that attributes the realized performance gap between the observed inventory policy and optimized reference policy to individual decisions. A numerical experiment on a simulated finite-horizon periodic-review inventory system with stochastic demand and lead time is conducted to illustrate the basic idea of the method. Compared to traditional methods, the proposed approach directly explains a realized benchmark-relative performance difference and is applicable to integer-constrained, non-differentiable, and simulation-based inventory systems. It enables transparent inventory management performance evaluation and effective root-cause analysis.

Keywords: inventory management; Shapley value; counterfactual decomposition; performance attribution

1. Introduction

Inventory management systems are typically evaluated using aggregate performance metrics such as service level, stockout frequency, fill rate, average inventory, or composite cost functions [1]. In practice, firms assess the quality of an ordering policy over a finite horizon by examining summary indicators that balance service performance and inventory efficiency. While such evaluation metrics are informative at the system level, they provide limited insight into which specific order decisions contributed most to performance deviations from a desired benchmark. To understand the underlying drivers of inventory performance during a given period, it is necessary to conduct retrospective analysis and quantify the contribution of past decisions to realized outcomes.

Such a task, although practically important, is analytically challenging. For example, in periodic-review systems with stochastic demand and lead time, order decisions are made at discrete epochs and propagate dynamically through future inventory trajectories. Because inventory evolves recursively, and current stock depends on prior stock levels, arrivals, and realized demand, the impact of a single decision is inherently intertemporal. A suboptimal decision at one review epoch may influence multiple downstream periods through altered stock levels, stockout occurrences, and holding patterns. Consequently, it is difficult to attribute overall performance to individual decisions in a principled manner.

Several quantitative approaches exist for analyzing the impact of inventory management decisions. For example, gradient-based sensitivity analysis evaluates local marginal effects [2], while variance-based global sensitivity methods decompose output variability across input factors [3,4]. Although valuable, as discussed in detail later, these methods have limitations and are not suitable for retrospective reason decomposition.

 doi:10.20944/preprints202603.2499.v2

To address these limitations, this paper proposes a novel and broadly applicable decomposition framework that attributes the gap between an observed inventory performance metric and an ideal benchmark to individual decisions. The proposed method is inspired by the Shapley value from cooperative game theory [5], which measures the average marginal contribution of a player across all possible coalitions. In our setting, the "players" are decision variables. For each decision, we compute its average marginal effect on the evaluation metric by comparing counterfactual policies in which subsets of other decisions are replaced by their ideal counterparts. This procedure yields an attribution measure that captures, in expectation, how much the evaluation metric would improve if a particular order were aligned with its ideal value, while accounting for interaction effects among decisions.

Beyond the core decomposition idea, several components of the proposed framework are designed to make the method applicable to realistic inventory management settings. First, the benchmark is defined using feasible optimized reference policies rather than unrestricted or purely theoretical solutions. When multiple high-performing reference policies are available, the framework can compute attributions relative to each reference policy and aggregate them using distance-based weights, reducing sensitivity to an arbitrary benchmark choice. Second, for large-scale simulation-based systems, the algorithm can incorporate a surrogate model to approximate the inventory evaluation function and reduce repeated simulation costs. Third, because exact Shapley computation becomes infeasible as the number of decision epochs increases, the framework uses a Monte Carlo estimator and can employ more efficient permutation-sampling strategies, such as antithetic sampling, to improve computational efficiency and numerical stability.

To illustrate the applicability of the decomposition method, we consider a finite-horizon multi-location periodic-review inventory system with stochastic demand and lead time. The system is evaluated using a composite metric that combines the proportion of periods experiencing stockouts and normalized average inventory. The decision variables are the order quantity at each ordering epoch. Applying the proposed method, we quantitatively calculate the contribution of each order decision that leads to the gap of observed inventory evaluation results and the results under optimized reference policy.

The contribution of this work is threefold. First, we formulate a Shapley-based counterfactual decomposition framework that attributes a realized benchmark-relative inventory performance gap to individual order decisions, treating decision epochs as the units of attribution rather than explaining local sensitivity, output variance, or model predictions. Second, we develop the framework into an operationally applicable algorithm by incorporating feasible optimized reference policies, feasibility-preserving counterfactual replacement, distance-based benchmark selection, multi-reference attribution averaging, permutation-based Monte Carlo estimation, antithetic sampling, and optional surrogate-assisted evaluation for computationally expensive simulations. Third, we demonstrate the method in a multi-location periodic-review inventory system with stochastic demand and variable lead times, showing how the framework identifies the location-epoch decisions that contribute most to realized performance loss and supports transparent retrospective inventory diagnosis.

One thing worth noting is that this paper does not seek to develop a new Shapley value solution concept. Rather, its contribution is to formulate a Shapley-based counterfactual decomposition framework tailored to retrospective inventory performance diagnosis. In contrast to existing inventory-game applications of the Shapley value, where the players are typically firms, retailers, or supply-chain members and the objective is to allocate cooperative savings or profits, the present framework treats order decision epochs as the units of attribution. In contrast to SHAP-style model explanation, where the objective is to explain the prediction of a trained black-box model, our objective is to decompose the realized inventory performance gap between an observed policy and an optimized benchmark policy.

The remainder of the paper is organized as follows. Section 2 does a literature review. Section 3 describes the existing traditional approaches. Section 4 explains the algorithm detail. Section 5 presents computational experiments. Section 6 discusses managerial implications, and Section 7 concludes.

 doi:10.20944/preprints202603.2499.v2

2. Literature Review

This paper builds on three streams of literature: (i) cooperative game theory and Shapley-value-based allocation in inventory systems, (ii) decomposition and sensitivity analysis for complex simulation models, and (iii) Shapley-based attribution methods used for explaining the output of black-box functions.

2.1. Cooperative Game Theory in Inventory Management

A substantial body of work applies cooperative game theory to inventory systems, primarily to allocate the gains from cooperation (e.g., risk pooling, inventory centralization, and joint replenishment) among participating agents. For example, [6] uses the Shapley value to allocate savings generated by inventory pooling in multi-retailer supply chains, showing that Shapley-based allocations can coordinate the supply chain and yield individually rational outcomes. [7] further develops this idea in the context of inventory centralization, using Shapley-based allocations to distribute profits in centralized systems.

Beyond these specific models, surveys and reviews synthesize the broader class of *inventory games* and highlight the role of solution concepts such as the core and the Shapley value in cost/profit allocation problems ([8]). Related research on deterministic multi-period replenishment also formulates cooperative games for lot-sizing and joint ordering ([9]).

2.2. Sensitivity Analysis and Variance-Based Decomposition

A second stream explores the decomposition of model outputs via sensitivity analysis. Local (derivative or finite-difference) sensitivity methods quantify marginal responsiveness near a baseline configuration but are often limited in non-smooth or integer-constrained systems. Global sensitivity analysis addresses this by apportioning output uncertainty to input factors, typically via variance-based indices. [3] provides a foundational treatment of global sensitivity indices and Monte Carlo estimators for nonlinear models. [4,10] develop practical estimators and best practices for computing total effect indices, emphasizing interaction effects and computational efficiency, and provide a comprehensive synthesis in a widely used monograph.

These approaches are valuable for understanding how *uncertainty* in inputs drives *variance* in outputs, but they generally do not answer the counterfactual question at the core of our study: given a realized underperformance relative to a benchmark, *which particular order decisions contributed most to the observed gap*? Our decomposition targets the realized performance difference $y(\mathbf{x}^*) - y(\mathbf{x})$ rather than variance, and it does so without requiring distributional assumptions over the decision variables.

2.3. Shapley-based Attribution for Explaining Complex Functions

A third, adjacent stream uses Shapley values for attributing the output of a complex function to its inputs, most prominently in interpretable machine learning. [11] propose SHAP, a unified framework that uses Shapley values to assign feature attributions for individual predictions and discusses desirable axioms such as consistency. The theoretical origin of this attribution principle traces back to Shapley's seminal axiomatization of a value for n-person games ([5]). Our method is conceptually aligned with Shapley-based attribution in that it computes average marginal contributions under coalitions of "corrected" inputs.

3. Existing Approaches

Various methodologies have been proposed in the inventory and operations literature to analyze the drivers of system performance. However, these approaches differ fundamentally in what they decompose and how attribution is assigned. This section situates our proposed decomposition framework relative to existing methods.

 doi:10.20944/preprints202603.2499.v2

3.1. Domain Knowledge-Based Decomposition

In practice, a widely used approach to diagnosing inventory performance relies on domain knowledge and heuristic reasoning. Managers often analyze operational metrics such as order placement quality (e.g., whether orders are systematically under- or over-sized), demand forecast accuracy, and realized lead-time delays to infer the underlying causes of inventory outcomes. By examining these factors, it is possible to form a qualitative understanding of the drivers behind observed performance.

While such domain knowledge-based analysis is intuitive and operationally convenient, it has several limitations. First, it does not provide a principled quantitative framework for evaluating the relative importance of different factors. Multiple drivers may simultaneously influence inventory outcomes, and their effects are often interdependent, making it difficult to disentangle their individual contributions. Second, this approach typically relies on manually selected metrics and human interpretation. As a result, the conclusions may depend on subjective judgment and may vary across analysts. In complex systems with nonlinear dynamics and intertemporal interactions, such heuristic reasoning may overlook subtle but important effects. Finally, domain knowledge-based methods do not naturally support counterfactual analysis. While they can indicate potential causes, they do not quantify how much performance would improve if a particular factor were corrected.

3.2. Gradient-Based Analysis

A common approach in inventory control and optimization is local sensitivity analysis, where the impact of decision variables on system performance is evaluated through marginal effects [2,12]. Let

$$\mathbf{x} = (x_1, x_2, \ldots, x_p)$$

denote the vector of decisions, where x_j represents the decision quantity(assuming all decision variables are numerical) at decision epoch j, and let $y(\mathbf{x})$ denote the corresponding inventory performance metric.

In this setting, the sensitivity of performance with respect to decision x_j is typically approximated using derivatives or finite differences, such as

$$\frac{\partial y}{\partial x_j} \quad \text{or} \quad \frac{y(x_1, \ldots, x_j + \epsilon, \ldots, x_p) - y(\mathbf{x})}{\epsilon}, \tag{1}$$

where ϵ is a small perturbation. These quantities measure the marginal change in system performance resulting from an infinitesimal or small change in the decision variable x_j.

Such methods provide useful directional information and are widely used in continuous optimization to study the stability and local behavior of optimal solutions [2]. In particular, they can indicate whether increasing or decreasing a specific decision quantity is likely to improve performance in the neighborhood of The current policy.

However, several limitations arise when applying gradient-based analysis of inventory systems. First, these methods are inherently local: they describe the responsiveness of the system near a fixed operating point $\mathbf{x}$ and do not capture how decisions influence performance globally across the decision space. As a result, they are not suitable for explaining large deviations between two policies. Second, inventory systems often exhibit nonlinear and non-smooth dynamics. For example, lost-sales models involve max operators, inventory truncation at zero, and discrete event transitions . In such settings, the mapping from decisions to performance may not be differentiable, and derivatives may be ill-defined or unstable. Third, decision variables are frequently non-numeric or constrained in practice, which further limits the applicability of differential methods. Finally, gradient-based methods do not capture interaction effects among decisions. In dynamic inventory systems, the impact of one decision may depend on other decisions through state propagation over time. While higher-order derivatives could, in principle, account for such interactions, they are rarely tractable in simulation-based or high-dimensional settings.

 doi:10.20944/preprints202603.2499.v2

3.3. Variance-Based Decomposition and Surrogate-Model-Based Attribution

Another class of approaches for explaining inventory performance is global sensitivity analysis. Unlike gradient-based methods, which characterize local responsiveness near a fixed operating point, global sensitivity methods study how variation in the input decisions contributes to variation in the output over a broader decision space [3,4]. In variance-based sensitivity analysis, the objective is not to decompose a realized gap between two policies, but rather to decompose the output variance $\mathrm{Var}(y(\mathbf{x}))$ into components associated with individual decision variables and their interactions.

A standard formulation is the functional ANOVA decomposition, under which the model output is represented as

$$y(\mathbf{x}) = f_0 + \sum_{j=1}^{p} f_j(x_j) + \sum_{i<j} f_{ij}(x_i, x_j) + \cdots + f_{1,\ldots,p}(x_1, \ldots, x_p), \tag{2}$$

where f_0 is a constant term, $f_j(x_j)$ captures the main effect of decision x_j, and higher-order terms represent interaction effects among decisions [4]. This decomposition induces a variance decomposition of the form

$$\mathrm{Var}(y) = \sum_{j=1}^{p} V_j + \sum_{i<j} V_{ij} + \cdots, \tag{3}$$

where $V_j = \mathrm{Var}\big(\mathbb{E}[y \mid x_j]\big)$ is the main-effect contribution of decision x_j, and V_{ij} denotes the pairwise interaction contribution between x_i and x_j [3,4]. The corresponding first-order Sobol index is $S_j = \frac{V_j}{\mathrm{Var}(y)}$, which measures the share of output variance explained by the decision x_j alone [3]. In principle, such quantities can be estimated by generating many decision combinations, simulating the inventory system under those combinations, and applying Monte Carlo estimators [3,4].

When the inventory simulator is computationally expensive or the decision space is high-dimensional, one may instead fit a surrogate model $\hat{y}(\mathbf{x}) \approx y(\mathbf{x})$. for example using a neural network, and then studying variable importance through the surrogate. In that case, one may use variance-based importance measures or feature-attribution tools such as SHAP values to quantify how the surrogate prediction depends on each input variable [11]. This strategy is attractive because it can reduce repeated simulation costs and leverage modern predictive models.

Despite their usefulness, these methods have several limitations for the retrospective decomposition problem considered in this paper. First, variance-based methods explain *performance variability* over a hypothetical distribution of decisions, rather than the realized performance difference between an observed policy and a benchmark policy. Second, these methods require the specification of a sampling distribution over decision variables. In practical inventory problems, this is not straightforward, because the decision space may be constrained and the decisions may be dependent. For example, later order quantities may be limited by earlier replenishment choices, budget constraints, or policy rules. Classical functional ANOVA and Sobol decompositions are most naturally formulated under independent inputs, and dependent inputs complicate both the interpretation and estimation of variance-based effects [4]. If simulated decision combinations do not respect the actual dependence structure, the resulting importance measures may be misleading. Third, the surrogate-model approach introduces an additional layer of approximation error. Reliable attribution then depends not only on the decomposition method itself, but also on the predictive accuracy of the fitted model. In high-dimensional settings, this may require a large number of simulated decision combinations for training and validation, which can be computationally demanding. Moreover, if the sampled combinations are infeasible or operationally implausible, the surrogate may learn relationships that do not reflect the true decision environment.

 doi:10.20944/preprints202603.2499.v2

4. Algorithm Details

This section explains the algorithm in detail. Each sub-section discusses one component of the algorithm. Table 1 gives an overall view of the algorithm, while Algorithm 1 outlines the detailed pseudo code.

Table 1. How the proposed framework addresses limitations of traditional approaches.

Approach	How the proposed framework addresses the limitation
Domain-knowledge-based diagnosis	Provides a formal quantitative attribution value for each decision epoch, reducing reliance on subjective managerial interpretation.
Gradient-based analysis	Does not require differentiability or continuous decision variables, making it applicable to integer-constrained and simulation-based inventory systems.
Variance-based decomposition	Decomposes the realized benchmark-relative gap $y(x^*) - y(x^{obs})$, rather than explaining output variance over a hypothetical input distribution.
Surrogate-model-based attribution	Can directly evaluate feasible counterfactual policies through the inventory system, avoiding attribution that depends solely on surrogate-model accuracy.

4.1. Decomposition Methodology

Let $\mathbf{x}$ denote the observed decision vector and let $\mathbf{x}^*$ denotes an reference optimized decision vector obtained via numerical optimization. Define

$$y = y(\mathbf{x}), \qquad y^* = y(\mathbf{x}^*), \tag{4}$$

and the performance gap

$$\Delta = y^* - y. \tag{5}$$

Our objective is to decompose Δ into contributions associated with each decision variable x_j.

Let $N = \{1, 2, \ldots, p\}$ denote the index set of decisions. For any subset $A \subseteq N$, define a hybrid decision vector

$$\mathbf{x}^{(A)} = \begin{cases} x_j^*, & \text{if } j \in A, \\ x_j, & \text{if } j \notin A. \end{cases} \tag{6}$$

Thus, decisions in A are replaced by their optimized reference counterparts.

Define the value function

$$v(A) = y(\mathbf{x}^{(A)}). \tag{7}$$

Observe that

$$v(\varnothing) = y(\mathbf{x}), \qquad v(N) = y(\mathbf{x}^*). \tag{8}$$

The decomposition problem becomes attributing

$$v(N) - v(\varnothing) \tag{9}$$

across the p decisions.

For decision $j \in N$, define its marginal contribution relative to subset $A \subseteq N \setminus \{j\}$ as

$$\Delta_j(A) = v(A \cup \{j\}) - v(A). \tag{10}$$

 doi:10.20944/preprints202603.2499.v2

The canonical Shapley value for decision j is

$$\phi_j = \sum_{A \subseteq N \setminus \{j\}} \frac{|A|!(p - |A| - 1)!}{p!} [v(A \cup \{j\}) - v(A)]. \tag{11}$$

The Shapley value satisfies the efficiency property

$$\sum_{j=1}^{p} \phi_j = v(N) - v(\varnothing), \tag{12}$$

ensuring that the total performance gap is fully allocated.

4.1.1. Properties of the Decomposition

The proposed decomposition inherits key properties from the Shapley value in cooperative game theory. For completeness, we state these properties in the context of our formulation.

Proposition 1. *Let $v(A) = y(\mathbf{x}^{(A)})$ be the value function defined over subsets of decision indices. The Shapley-based contribution ϕ_j satisfies:*

- ***Efficiency: Contributions sum to the total gap***

$$\sum_{j=1}^{p} \phi_j = v(N) - v(\varnothing).$$

- ***Symmetry: If two decisions have identical effects in all coalitions, they receive identical contributions.*** *If for two decisions i and j,*

$$v(A \cup \{i\}) = v(A \cup \{j\}) \quad \forall A \subseteq N \setminus \{i, j\},$$

then $\phi_i = \phi_j$.

- ***Null player: If correcting a decision never changes performance, its contribution is zero.*** *If*

$$v(A \cup \{j\}) = v(A) \quad \forall A \subseteq N \setminus \{j\},$$

then $\phi_j = 0$.

These properties follow directly from the linearity and symmetry of the Shapley value construction. The efficiency property arises from telescoping marginal contributions across permutations. Symmetry follows from identical marginal effects across all coalitions, and the null player property holds when marginal contributions are zero in all contexts. See [5] for the original derivation.

Algorithm 1 Surrogate-Assisted Feasibility-Preserving Multi-Benchmark Monte Carlo Shapley Decomposition

Require: Observed policy x^{obs}; feasible set $\mathcal{X}$; operational policy class $\mathcal{X}_{policy}$; inventory simulator $y(\cdot)$; optimized reference candidate set $\mathcal{R}$; distance threshold ϵ; number of retained benchmarks K; number of Monte Carlo permutations M; repair/projection operator $\Pi_{\mathcal{X}}(\cdot)$; distance function $d(\cdot, \cdot)$; distance-weighting constant $\delta > 0$; surrogate validation tolerance τ_{sur}; surrogate retraining interval B; simulator audit probability q_{sim}.

Ensure: Decision-level attribution values $\bar{\phi}_1, \ldots, \bar{\phi}_p$ and improvement contributions $I_1, \ldots, I_p$.

1: **Step 1: Generate feasible optimized reference candidates.**

2: Solve

$$\min_{x \in \mathcal{X}_{policy}} y(x), \qquad \mathcal{X}_{policy} \subseteq \mathcal{X}.$$

3: Store feasible optimized reference candidates:

$$\mathcal{R} = \{x^{*,1}, x^{*,2}, \ldots, x^{*,H}\}.$$

4: Store all simulator-evaluated candidate policies and outcomes in the initial dataset:

$$\mathcal{D} = \{(x^{(r)}, y(x^{(r)}))\}_{r=1}^{R_{opt}}.$$

 doi:10.20944/preprints202603.2499.v2

Algorithm 1 *Cont.*

Step 2: Select benchmark policy or benchmark set by distance.
for $h = 1, \ldots, H$ **do**
  Compute

$$d_h = d(x^{*,h}, x^{obs}).$$

end for
Construct

$$\mathcal{B}_\epsilon = \left\{ x^{*,h} \in \mathcal{R} : d(x^{*,h}, x^{obs}) \leq \epsilon \right\}.$$

if $\mathcal{B}_\epsilon = \emptyset$ **then**
  Select

$$x^* = \arg\min_{x^{*,h} \in \mathcal{R}} d(x^{*,h}, x^{obs}).$$

  Set $\mathcal{B} = \{x^*\}$.
else if single-benchmark decomposition is used **then**
  Select1

$$x^* = \arg\min_{x^{*,h} \in \mathcal{B}_\epsilon} d(x^{*,h}, x^{obs}).$$

  Set $\mathcal{B} = \{x^*\}$.
else
  Retain up to K policies from $\mathcal{B}_\epsilon$ with the smallest distances to x^{obs}.
  Denote the retained benchmark set by

$$\mathcal{B} = \{x^{*,1}, x^{*,2}, \ldots, x^{*,K'}\}, \qquad K' = \min\{K, |\mathcal{B}_\epsilon|\}.$$

end if
Step 3: Train and validate initial surrogate.
Split $\mathcal{D}$ into training and validation sets.
Train a surrogate model $\hat{y}(x; \theta)$ using the training set.
Compute validation error E_{val}, such as MAE_{val} or $RMSE_{val}$.
if $E_{val} \leq \tau_{sur}$ **then**
  Set surrogate status to active.
else
  Set surrogate status to inactive.
end if
Step 4: Initialize attribution values.
for each benchmark policy $x^{*,k} \in \mathcal{B}$ **do**
  for each decision epoch $j = 1, \ldots, p$ **do**
    Set $\hat{\phi}_j^{(k)} = 0$.
  end for
end for

 doi:10.20944/preprints202603.2499.v2

Algorithm 1 *Cont.*

Step 5: Monte Carlo Shapley estimation with surrogate-assisted coalition evaluation.
Set simulator evaluation counter $C = 0$.
for each benchmark policy $x^{*,k} \in \mathcal{B}$ **do**
  for $m = 1, \ldots, M$ **do**
    Sample a random permutation π_m of decision epochs $\{1, \ldots, p\}$.
    Set current policy $x^{cur} = x^{obs}$.
    Repair current policy:

$$\tilde{x}^{cur} = \Pi_{\mathcal{X}}(x^{cur}).$$

    Evaluate current value:

$$v^{cur} = \text{Evaluate}(\tilde{x}^{cur}).$$

    for each decision epoch j in the order π_m **do**
      Replace the observed decision with the benchmark decision:

$$x_j^{cur} \leftarrow x_j^{*,k}.$$

      Repair the hybrid policy:

$$\tilde{x}^{new} = \Pi_{\mathcal{X}}(x^{cur}).$$

      Evaluate the new coalition value:

$$v^{new} = \text{Evaluate}(\tilde{x}^{new}).$$

      Record the marginal contribution:

$$\Delta_j^{(m,k)} = v^{new} - v^{cur}.$$

      Update:

$$v^{cur} \leftarrow v^{new}, \qquad x^{cur} \leftarrow \tilde{x}^{new}.$$

    end for
    if C is positive and C is a multiple of B **then**
      Retrain surrogate $\hat{y}(x; \theta)$ using the updated dataset $\mathcal{D}$.
      Recompute validation error E_{val} on newly simulated validation policies.
      if $E_{val} \leq \tau_{sur}$ **then**
        Set surrogate status to active.
      else
        Set surrogate status to inactive.
      end if
    end if
  end for
  for each decision epoch $j = 1, \ldots, p$ **do**
    Compute

$$\hat{\phi}_j^{(k)} = \frac{1}{M} \sum_{m=1}^{M} \Delta_j^{(m,k)}.$$

    Compute Monte Carlo standard error

$$SE_j^{(k)} = \frac{s_j^{(k)}}{\sqrt{M}}.$$

  end for
end for
Step 6: Average attribution values across benchmarks.
for each benchmark policy $x^{*,k} \in \mathcal{B}$ **do**
  Compute distance-based weight

$$w_k = \frac{[d(x^{*,k}, x^{obs}) + \delta]^{-1}}{\sum_{\ell=1}^{|\mathcal{B}|} [d(x^{*,\ell}, x^{obs}) + \delta]^{-1}}.$$

end for
for each decision epoch $j = 1, \ldots, p$ **do**
  Compute

$$\bar{\phi}_j = \sum_{k=1}^{|\mathcal{B}|} w_k \hat{\phi}_j^{(k)}.$$

  Compute improvement contribution:

$$I_j = -\bar{\phi}_j.$$

end for
return $\{(\bar{\phi}_j, I_j)\}_{j=1}^{p}$.

4.1.2. Intuition Behind the Shapley-Based Contribution

The Shapley-style contribution used in our framework can be interpreted as the expected marginal improvement obtained when a particular decision is aligned with its 'ideal' value, averaged across all possible interaction contexts.

In a dynamic decision system like inventory management, the impact of a single decision depends on other decisions made before and after it. For example, correcting an early under-order may have little effect if subsequent orders compensate for the shortfall. Conversely, the same correction may

 doi:10.20944/preprints202603.2499.v2

substantially reduce stockouts if later decisions are also aligned with the benchmark policy. Thus, the marginal effect of correcting decision j is inherently context-dependent.

Formally, the marginal contribution of decision j depends on the subset A of other decisions already corrected:

$$\Delta_j(A) = v(A \cup \{j\}) - v(A). \tag{13}$$

Different subsets A may produce different marginal effects due to interaction among decisions.

The quantity ϕ_j can be interpreted as:The expected reduction in the performance gap if order decision j were aligned with the optimized reference policy, accounting for interaction with other decisions.

If ϕ_j is large in magnitude, decision j plays a substantial role in explaining the gap between the observed and optimized reference policies. If ϕ_j is close to zero, correcting that decision alone has a limited impact once interactions are averaged.

One may view the optimized reference policy as a fully corrected system. Starting from the observed policy, imagine gradually correcting decisions in random order. Each time a decision is corrected, the performance improves by some amount. The Shapley contribution assigns to each decision its average improvement across all possible correction sequences.

4.1.3. Positioning Relative to Traditional Decomposition Approaches

Section 3 discussed several existing approaches for analyzing inventory performance, including domain-knowledge-based diagnosis, gradient-based sensitivity analysis, variance-based decomposition, and surrogate-model-based attribution. Table 1 summarizes the central methodological distinction of the proposed framework. Overall, the proposed method is better suited to the retrospective reason-decomposition problem studied in this paper. It is benchmark- relative, interaction-aware, globally interpretable, and applicable to nonlinear, non-differentiable, and integer-constrained inventory systems. These properties make it more informative than domain knowledge-based diagnosis, more globally relevant than gradient-based analysis, and more managerially meaningful than variance-based decomposition for explaining realized inventory performance.

4.1.4. Comparison with Expectation-Based Attribution

To further explain the advantages of the proposed decomposition method, we comparee the method with an alternative, more traditional perspective for defining the contribution of each decision variable. Such a traditional approach is based on comparing the realized performance with an expected baseline. Specifically, one may define the contribution of decision x_j as

$$\phi'_j = y(\mathbf{x}) - \mathbb{E}_{x_{-j}}\left[y(x_j, x_{-j})\right], \tag{14}$$

where x_{-j} denotes all decision variables except x_j, and the expectation is taken over a distribution of possible values for the remaining decisions.

While this formulation is conceptually appealing in some contexts, it is less suitable for the inventory performance decomposition problem considered in this paper.

First, the two approaches differ fundamentally in their reference points. The expectation-based formulation attributes the difference between the realized performance $y(\mathbf{x})$ and an average performance across hypothetical decision combinations. In contrast, the proposed method attributes the difference between the realized performance and an optimized benchmark. From a managerial perspective, the benchmark-relative comparison is more meaningful, as decision makers are typically concerned with how far the current policy deviates from an attainable optimal or target policy, rather than from an abstract average over all possible decisions.

Second, the expectation-based approach requires specifying a distribution over x_{-j} and evaluating the performance over all possible combinations of decision variables. In high-dimensional settings, this leads to significant computational challenges. Moreover, many such combinations may be operationally

 doi:10.20944/preprints202603.2499.v2

infeasible (e.g., violating capacity or policy constraints), which can distort the resulting attribution. In contrast, the proposed decomposition considers a structured set of counterfactual scenarios in which decisions are transitioned from their observed values to their benchmark values. This reduces the computational burden and ensures that all evaluated combinations remain interpretable and relevant to the underlying decision problem.

4.1.5. Comparison with the Current Shapley Attribution Method

Although the proposed framework is inspired by the Shapley value and is conceptually related to SHAP-style attribution, it differs from these methods in the object of attribution, the construction of the value function, and the interpretation of the resulting contributions. In classical cooperative-game applications in inventory management, Shapley values are commonly used to allocate the gains from cooperation, such as inventory pooling or centralized replenishment, among multiple firms or retailers. The unit of allocation is therefore an economic agent. In the present study, by contrast, the unit of attribution is an inventory decision epoch within a single dynamic ordering process.

The proposed method also differs from SHAP-style feature attribution. SHAP explains the output of a predictive model by assigning contributions to input features, usually relative to an expected model output. In our framework, the function being explained is not a black-box prediction model but an inventory evaluation function generated by simulation or optimization. The baseline is not an unconditional expectation over feature values, but the realized observed policy. The reference point is not an average prediction, but an optimized benchmark policy. Therefore, the attribution answers a different question: which order decisions explain the realized performance gap between the observed policy and the benchmark policy?

This distinction is important for managerial inventory diagnosis. A manager evaluating a completed planning horizon is often less interested in explaining model predictions or output variance over hypothetical decision distributions than in identifying which past decisions contributed most to underperformance relative to an attainable benchmark. The proposed framework addresses this diagnostic question by constructing hybrid counterfactual policies and averaging each decision's marginal contribution across possible correction sequences.

4.2. Feasible and Robust Selection of Optimized Reference Policy

The proposed decomposition is based on comparing the observed policy with an optimized reference policy. Because the attribution results depend on this reference policy, the benchmark must be selected carefully. In this study, we use the following principles to construct and use optimized reference policies.

First, the optimized reference policy must be feasible. let $\mathcal{X}$ denote the feasible policy set. The feasible set includes all operational constraints, such as nonnegative integer order quantities, capacity limits, budget constraints, and other replenishment rules. Therefore, any optimized reference policy x^* must satisfy $x^* \in \mathcal{X}$.

This means that the optimized referenced is defeined as following:

$$x^* := argmin_{x \in \mathcal{X}} y(x) \tag{15}$$

This requirement ensures that the reference policy is not only numerically high-performing, but also operationally valid.

Second, there may be multiple optimized reference policies. In nonlinear or integer-constrained inventory systems, different feasible policies may achieve the same optimized evaluation value. Since the attribution result depends on the chosen reference policy, it is important to avoid selecting an arbitrary benchmark. In this study, we select optimized reference policies that are close to the observed policy.

Let

$$\mathcal{R} = \{x^{*,1}, x^{*,2}, \ldots, x^{*,H}\}$$

 doi:10.20944/preprints202603.2499.v2

denote the set of feasible optimized reference policies obtained from the optimization procedure. These policies have the same optimized evaluation value, or are treated as alternative optimized reference policies generated by the benchmark construction procedure. To select the benchmark, we measure the distance between each optimized reference policy and the observed policy:

$$d(x^{*,h}, x^{obs}), \qquad h = 1, \ldots, H.$$

A simple choice is the absolute deviation distance:

$$d(x^{*,h}, x^{obs}) = \sum_{j=1}^{p} |x_j^{*,h} - x_j^{obs}|. \tag{16}$$

Alternatively, one may use the normalized Euclidean distance:

$$d(x^{*,h}, x^{obs}) = \left[\sum_{j=1}^{p} \left(\frac{x_j^{*,h} - x_j^{obs}}{\max\{1, x_j^{obs}\}} \right)^{\lambda} \right]^{1/\lambda} \tag{17}$$

The primary benchmark can then be selected as the optimized reference policy that is closest to the observed policy:

$$x^* = \arg \min_{x^{*,h} \in \mathcal{R}} d(x^{*,h}, x^{obs}). \tag{18}$$

This rule selects the optimized reference policy that requires the smallest deviation from the observed policy. As a result, the decomposition attributes the performance gap relative to the most conservative optimized benchmark.

Alternatively, instead of selecting only one benchmark, one may retain all optimized reference policies whose distance from the observed policy is within a pre-specified threshold:

$$\mathcal{B}_\epsilon = \left\{ x^{*,h} \in \mathcal{R} : d(x^{*,h}, x^{obs}) \leq \epsilon \right\}. \tag{19}$$

Here, ϵ controls how far an optimized reference policy is allowed to be from the observed policy. A smaller ϵ restricts the decomposition to reference policies that are very close to the observed policy, while a larger ϵ allows a broader set of optimized reference policies to be included.

4.3. Feasible Counterfactual Policy

Third, when decomposing the performance difference between the observed policy and the optimized reference policy, every counterfactual policy generated during the permutation procedure should remain feasible. For a subset of decision epochs $A \subseteq N = \{1, \ldots, p\}$, the basic counterfactual policy is define in Equation 6. However, this hybrid policy may violate operational constraints. Therefore, before evaluating its performance, we apply a feasibility repair or projection operator

$$\Pi_{\mathcal{X}} : \mathbb{R}^p \to \mathcal{X}.$$

The feasible counterfactual policy is defined as

$$\tilde{x}^{(A)} = \Pi_{\mathcal{X}} \left(x^{(A)} \right). \tag{20}$$

The coalition value function is then computed as

$$v(A) = y\left(\tilde{x}^{(A)} \right) = y\left(\Pi_{\mathcal{X}} \left(x^{(A)} \right) \right). \tag{21}$$

 doi:10.20944/preprints202603.2499.v2

Therefore, the marginal contribution of decision epoch j is

$$\Delta_j(A) = v(A \cup \{j\}) - v(A),$$

where both $v(A \cup \{j\})$ and $v(A)$ are evaluated using feasible counterfactual policies. This guarantees that the Shapley decomposition is based only on operationally valid policies.

4.4. Monte Carlo Method for Estimating ϕ

Exact computation requires evaluating 2^p coalitions, which becomes computationally infeasible for large p. We therefore employ a Monte Carlo permutation-based estimator.

Let π be a random permutation of N. For each permutation:

1. Start from baseline $\mathbf{x}$.
2. Sequentially replace decisions by their ideal values following π.
3. Record the marginal effect when decision j is switched:

$$\Delta_j^{(\pi)} = v(S_\pi(j) \cup \{j\}) - v(S_\pi(j)), \tag{22}$$

where $S_\pi(j)$ is the set of decisions preceding j in permutation π.

The Monte Carlo Shapley estimator is

$$\hat{\phi}_j = \frac{1}{M} \sum_{m=1}^{M} \Delta_j^{(\pi_m)}, \tag{23}$$

where M is the number of sampled permutations. As $M \to \infty$, $\hat{\phi}_j$ converges to ϕ_j.

As an alternative decomposition, we define

$$\tilde{\phi}_j = \mathbb{E}_{A \subseteq N \setminus \{j\}}[v(A \cup \{j\}) - v(A)], \tag{24}$$

where each subset is assigned equal probability. In practice, we approximate this expectation via Monte Carlo sampling of subsets.

Although the permutation-based Monte Carlo estimator provides a practical way to approximate Shapley contributions, its accuracy depends on the number and quality of sampled permutations. Standard random permutation sampling may converge slowly, especially when the marginal contribution of a decision varies substantially across coalition contexts. Therefore, variance-reduction techniques can be used to improve the efficiency and reliability of the estimator. In our algorithm, we consider sampling methods such as antithetic permutation sampling and stratified permutation sampling. The explanation of these two techniques are outlined in the Appendix.

4.5. Surrogate-Assisted Coalition Evaluation

In large-scale simulation-based inventory systems, repeatedly evaluating $y(x)$ during Monte Carlo Shapley estimation can be computationally expensive. To reduce this cost, the proposed algorithm can be extended with a surrogate-assisted evaluation module. The main idea is to use policy-performance pairs generated during benchmark optimization and permutation-based decomposition to train an approximation model

$$\hat{y}(x) \approx y(x).$$

During benchmark optimization, many candidate policies are simulated, producing $(x, y(x))$ pairs. During the Shapley permutation procedure, additional hybrid counterfactual policies are also simulated. These observations are accumulated into a dataset

$$\mathcal{D} = \{(x^{(r)}, y(x^{(r)}))\}_{r=1}^{R}.$$

A surrogate model, such as a neural network or gradient-boosting model, can then be trained on $\mathcal{D}$.

 doi:10.20944/preprints202603.2499.v2

The surrogate is used only when its validation accuracy is sufficiently high. Let $\mathcal{D}_{val}$ be a validation set of newly simulated counterfactual policies. The validation error can be measured by

$$MAE_{val} = \frac{1}{|\mathcal{D}_{val}|} \sum_{(x,y(x))\in\mathcal{D}_{val}} |\hat{y}(x) - y(x)|.$$

If

$$MAE_{val} \leq \tau_{sur},$$

then the algorithm may use

$$\hat{v}(A) = \hat{y}\Big(\Pi_{\mathcal{X}}(x^{(A)})\Big)$$

instead of

$$v(A) = y\Big(\Pi_{\mathcal{X}}(x^{(A)})\Big)$$

for some coalition evaluations. After every B new simulator evaluations, the surrogate is retrained and revalidated. Even when the surrogate is active, a fraction of counterfactual policies should still be evaluated by the original simulator to monitor surrogate accuracy and update the training dataset.

4.6. Averaging of Attributions under Multiple Optimized Reference Policies

If multiple optimized reference policies are retained, the final attribution can be obtained by averaging the attribution values across these benchmarks. Suppose

$$\mathcal{B}_\epsilon = \{x^{*,1}, x^{*,2}, \ldots, x^{*,K}\}$$

is the selected set of feasible optimized reference policies satisfying

$$d(x^{*,k}, x^{obs}) \leq \epsilon, \qquad k = 1, \ldots, K.$$

For each reference policy $x^{*,k}$, we compute the Shapley attribution value $\phi_j^{(k)}$ for decision epoch j. A simple average attribution is

$$\bar{\phi}_j = \frac{1}{K} \sum_{k=1}^{K} \phi_j^{(k)}. \tag{25}$$

Alternatively, one may use a distance-weighted average, giving larger weight to optimized reference policies that are closer to the observed policy. Define

$$w_k = \frac{\Big[d(x^{*,k}, x^{obs}) + \delta\Big]^{-1}}{\sum_{\ell=1}^{K} \Big[d(x^{*,\ell}, x^{obs}) + \delta\Big]^{-1}} \tag{26}$$

where $\delta > 0$ is a small constant used to avoid division by zero. The distance-weighted attribution is then

$$\bar{\phi}_j^{w} = \sum_{k=1}^{K} w_k \phi_j^{(k)}. \tag{27}$$

Under this weighting rule, optimized reference policies closer to the observed policy receive larger weights. This reduces sensitivity to any single optimized reference policy and makes the final attribution more conservative.

 doi:10.20944/preprints202603.2499.v2

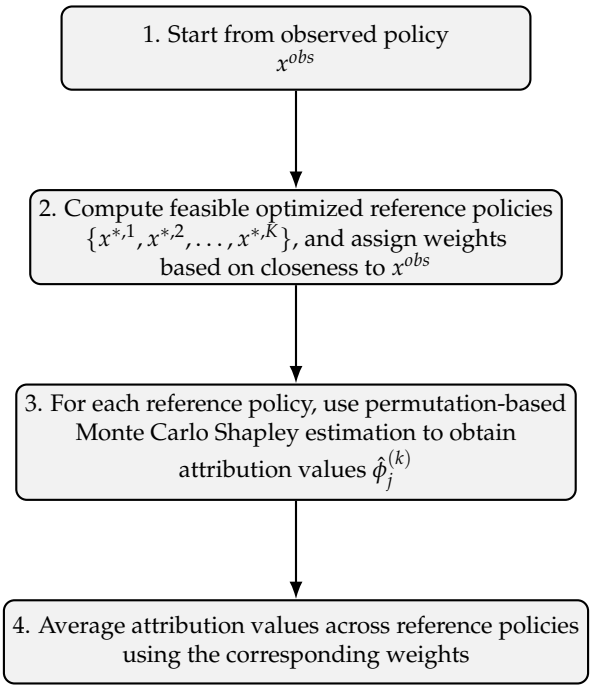

Figure 1. Brief workflow of the proposed decomposition framework. Starting from the observed policy, the method computes feasible optimized reference policies, estimates attribution values under each reference policy using permutation-based Monte Carlo Shapley decomposition, and then averages the attributions across reference policies using distance-based weights.

5. Numerical Experiment

This section illustrates the proposed decomposition framework using a multi-location inventory system with stochastic demand and variable lead times. The purpose of this experiment is to demonstrate that the proposed framework can be applied to a more general inventory setting while still providing interpretable decision-level attribution.

5.1. Multi-Location Inventory Setting

Consider a finite-horizon inventory system with L locations and T periods. Orders are placed every r periods. Let

$$\mathcal{J} = \{1, \ldots, p\}$$

denote the set of order decision epochs, where $p = T/r$. The decision variable is

$$x_{\ell,j}, \qquad \ell = 1, \ldots, L, \quad j = 1, \ldots, p,$$

where $x_{\ell,j}$ denotes the order quantity for location ℓ at decision epoch j. The full decision vector is written as

$$x = \left(x_{1,1}, \ldots, x_{1,p}, x_{2,1}, \ldots, x_{2,p}, \ldots, x_{L,1}, \ldots, x_{L,p}\right).$$

Thus, each Shapley "player" is a location-epoch ordering decision.

The feasible policy set $\mathcal{X}$ includes integer order constraints, lower and upper bounds, and a capacity constraint across locations at each decision epoch:

$$x_{\ell,j} \in \mathbb{Z}_{\geq 0}, \qquad q_{\min} \leq x_{\ell,j} \leq q_{\max},$$

and

$$\sum_{\ell=1}^{L} x_{\ell,j} \leq C_j, \qquad j = 1, \ldots, p.$$

 doi:10.20944/preprints202603.2499.v2

If a counterfactual hybrid policy violates these constraints, the policy is repaired using the feasibility projection operator $\Pi_{\mathcal{X}}(\cdot)$.

5.2. Realized Stochastic Demand and Lead-Time Environment

To reflect stochastic operational conditions while preserving the retrospective nature of the decomposition, demand and lead times are generated stochastically once and then treated as the realized historical environment. Let ω^{obs} denote this realized environment, including both demand and lead-time realizations. The evaluation metric is therefore written as

$$y(x) = g(x, \omega^{obs}), \tag{28}$$

where $g(x, \omega^{obs})$ is the realized inventory performance of policy x under the generated demand and lead-time trajectory.

For each location ℓ, demand is generated from a stochastic seasonal process:

$$D_{\ell,t} \sim \text{Poisson}(\mu_{\ell,t}), \quad \mu_{\ell,t} = b_\ell + a_\ell \sin\left(\frac{2\pi t}{s} + \varphi_\ell\right) \tag{29}$$

with b_ℓ representing the base demand level, a_ℓ the seasonal amplitude, s the seasonal cycle length, and φ_ℓ a location-specific phase shift. Lead times are also generated stochastically:

$$L_{\ell,j} \sim \text{DiscreteUniform}\{L_{\min}, \ldots, L_{\max}\} \tag{30}$$

After these stochastic quantities are generated, they are fixed for all policy evaluations. This design is consistent with the retrospective goal of the proposed framework: the objective is to explain the realized performance gap in a completed planning horizon, rather than to estimate expected long-run performance over many future scenarios.

5.3. Inventory Dynamics, Evaluation Metrics

Let $S_{\ell,t}$ denote the end-of-period inventory level at location ℓ in period t. Orders placed at decision epoch j arrive after the realized lead time $L_{\ell,j}$. Let $A_{\ell,t}(x)$ denote the realized arrival quantity at location ℓ in period t under policy x. The inventory dynamics are

$$S_{\ell,t} = \max\{0,\, S_{\ell,t-1} + A_{\ell,t}(x) - D_{\ell,t}\} \tag{31}$$

Stockout occurs when available inventory before demand fulfillment is less than realized demand. The inventory performance metric is a loss-type composite metric:

$$y(x) = w \cdot \text{OOS}(x) + (1 - w) \cdot \frac{\bar{S}(x)}{\bar{D}}, \tag{32}$$

where

$$\text{OOS}(x) = \frac{\text{number of location-periods with stockout}}{L \times T},$$

$$\bar{S}(x) = \frac{1}{LT} \sum_{\ell=1}^{L} \sum_{t=1}^{T} S_{\ell,t}, \quad \bar{D} = \frac{1}{LT} \sum_{\ell=1}^{L} \sum_{t=1}^{T} D_{\ell,t}.$$

The parameter $w \in [0, 1]$ controls the trade-off between stockout reduction and inventory efficiency. Since $y(x)$ is a loss-type metric, a smaller value indicates better inventory performance.

5.4. Searching for Optimized Reference Policies

The proposed decomposition compares the observed policy with one or more optimized reference policies. In the numerical experiment, these reference policies are generated using a heuristic search

 doi:10.20944/preprints202603.2499.v2

procedure. Therefore, they should be interpreted as high-performing feasible benchmarks rather than globally optimal policies.

The search starts from the observed policy x^{obs}. Candidate reference policies are created by perturbing the observed order quantities. The perturbation includes both a systematic upward adjustment and random noise. The upward adjustment allows candidate policies to reduce possible stockouts, while the random noise creates diversity across candidate policies. After each candidate policy is generated, it is passed through the feasibility repair operator $\Pi_{\mathcal{X}}(\cdot)$, which ensures that the policy satisfies integer constraints, order bounds, and capacity constraints.

Each feasible candidate policy is then evaluated using the same inventory performance metric $y(\cdot)$ used throughout the experiment. Since $y(\cdot)$ is a loss-type metric, a smaller value indicates better performance. After all candidates are evaluated, the best-performing feasible candidates are retained as the optimized reference candidate set.

When a single reference policy is required, the retained candidate closest to the observed policy is selected as x^*. When multiple reference policies are used, the retained candidates closest to the observed policy are selected and assigned weights based on their distance from x^{obs}. Attribution values are first computed separately under each selected reference policy and then averaged using these weights.

5.5. Experimental Parameters

Table 2 summarizes the main parameters used in the extended numerical experiment. Since the environment for this numerical experiment is not complicated, the inventory system outcome can be efficiently computed via simulation, we skipped the model-fitting component.

Table 2. Parameters for the extended multi-location numerical experiment.

Category	Parameter	Value
Inventory system	Planning horizon T	24 periods
Inventory system	Number of locations L	2
Inventory system	Review interval r	4 periods
Inventory system	Order epochs per location $p = T/r$	6
Inventory system	Total decision variables Lp	12
Inventory system	Initial inventory	20
Inventory system	Minimum order quantity $q_{\min}$	0
Inventory system	Maximum order quantity $q_{\max}$	80
Inventory system	Capacity per epoch C_j	130
Demand and lead time	Demand process	Stochastic seasonal Poisson
Demand and lead time	Lead-time range	1–3 periods
Evaluation metric	Stockout weight w	0.6
Reference generation	Number of candidate policies	80
Reference generation	Number of retained candidates	12
Benchmark averaging	Number of benchmarks K	3
Monte Carlo estimation	Number of permutations M	300

5.6. Monte Carlo Convergence

In this numerical experiment, we use antithetic permutation sampling to improve the efficiency of the Monte Carlo Shapley estimator.

Also, to evaluate the numerical reliability of the Monte Carlo estimator, the decomposition is repeated under different numbers of sampled permutations. For each M, we compute the mean and maximum Monte Carlo standard errors of the estimated attribution values.

The Monte Carlo convergence analysis evaluates the numerical stability of the estimated Shapley attribution values. Since the exact Shapley value requires evaluating all possible coalitions, the proposed method uses a permutation-based Monte Carlo estimator. For a fixed number of sam-

 doi:10.20944/preprints202603.2499.v2

pled permutations M, the attribution value for each decision is estimated by averaging its marginal contribution across the sampled permutations.

To quantify Monte Carlo uncertainty, we compute the sample variability of the marginal contributions obtained across permutations. For each decision j, the Monte Carlo standard error is then computed as

$$SE(\hat{\phi}_j) = \frac{s_j}{\sqrt{M}}, \tag{33}$$

where s_j is the sample standard deviation of $\Delta_j^{(1)}, \ldots, \Delta_j^{(M)}$. Therefore, the convergence analysis does not measure inventory-demand uncertainty; rather, it measures the numerical sampling uncertainty caused by approximating the Shapley value with a finite number of random permutations.

Figure 2 visualizes the Monte Carlo convergence pattern. In this numerical experiment, as As M approaches to 300, the standard errors decreases to a stable low level.

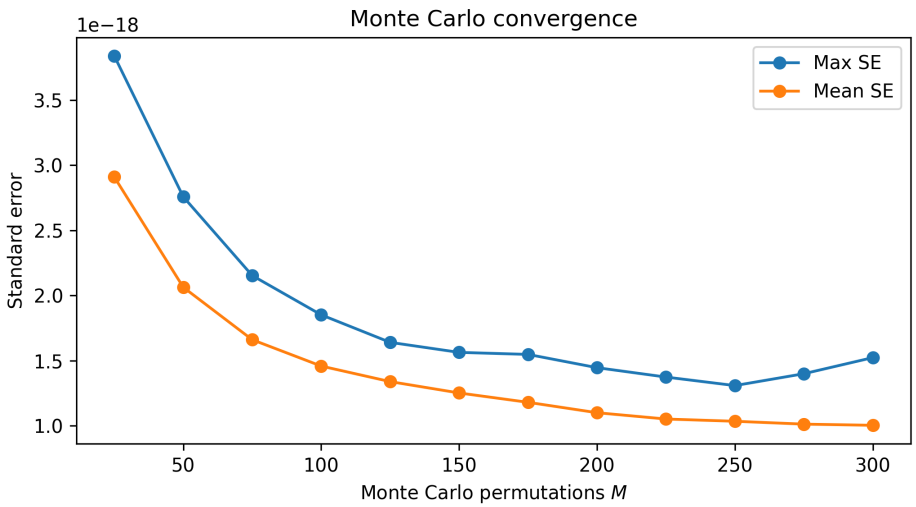

Figure 2. Monte Carlo convergence diagnostic. The mean and maximum standard errors of the estimated attribution values decrease as the number of sampled permutations increases, indicating improved numerical stability of the Shapley estimator.

5.7. Decision-Level Attribution Results

Figure 3 visualizes the location-epoch attribution results. Since the evaluation metric is a loss-type metric, larger positive values of I_j indicate that the corresponding observed location-epoch decision contributed more strongly to the performance loss relative to the optimized reference policies. Figure 3 indicates that in this specific numerical example, the current value of decision epoch L1-E1, L2-E3, L2-E4 is contributing most to the average improvement of the evaluation metric if they switch to their optimized reference counterparts. In contrast, L1-E4, L1-E5, L2-E1 on average makes the evaluation metric worse if they switch to their optimizd reference counterparts.

 doi:10.20944/preprints202603.2499.v2

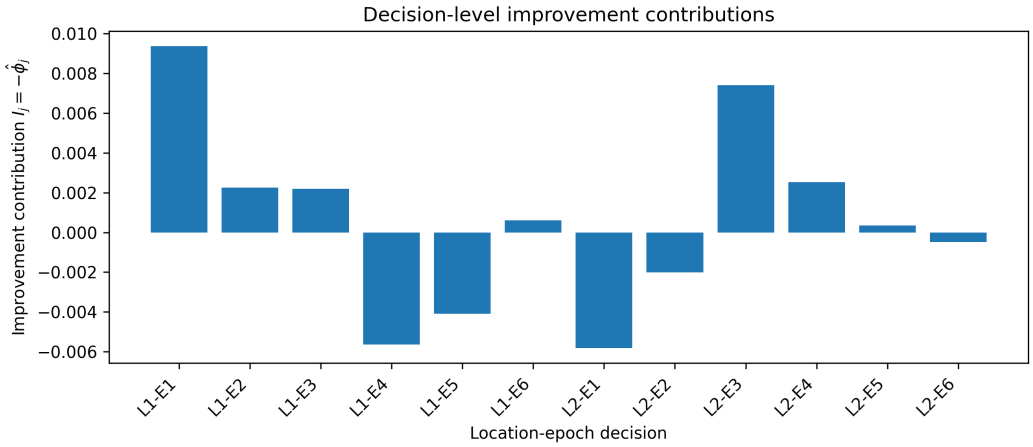

Figure 3. Location-epoch improvement contributions. Since the evaluation metric is minimized, the improvement contribution is reported as $I_j = -\bar{\phi}_j$, where a larger positive value indicates a larger expected reduction in the evaluation metric.

5.8. Sensitivity to Algorithm Parameters

Finally, we evaluate the sensitivity of the decomposition to algorithm parameters, including the distance-order parameter λ, the number of retained benchmark policies K, the number of Monte Carlo permutations M, and the averaging method.

Table 3 summarizes the sensitivity analysis, which shows that the algorithm yield similar important decision epochs across reasonable choices of λ, K, M, and the averaging scheme. The top and lowest contributors are in general consistent with the observation in section 5.7.

Table 3. Top and lowest contributors under different algorithm parameter settings.

λ	K	M	Weighting	Top 3 contributors	Lowest 3 contributors
1	1	100	Distance	[L2-E3, L1-E1, L2-E4]	[L1-E4, L2-E2, L2-E1]
1	1	100	Simple	[L2-E3, L1-E1, L2-E4]	[L1-E4, L2-E2, L2-E1]
1	1	300	Distance	[L2-E3, L1-E1, L2-E4]	[L1-E4, L2-E2, L2-E1]
1	1	300	Simple	[L2-E3, L1-E1, L2-E4]	[L1-E4, L2-E2, L2-E1]
1	3	100	Distance	[L1-E1, L2-E3, L2-E4]	[L1-E4, L2-E1, L1-E5]
1	3	100	Simple	[L1-E1, L2-E3, L2-E4]	[L1-E4, L2-E1, L1-E5]
1	3	300	Distance	[L1-E1, L2-E3, L2-E4]	[L1-E4, L2-E1, L1-E5]
1	3	300	Simple	[L1-E1, L2-E3, L2-E4]	[L1-E4, L2-E1, L1-E5]
2	1	100	Distance	[L2-E3, L1-E1, L1-E2]	[L2-E1, L2-E6, L1-E5]
2	1	100	Simple	[L2-E3, L1-E1, L1-E2]	[L2-E1, L2-E6, L1-E5]
2	1	300	Distance	[L2-E3, L1-E1, L1-E2]	[L2-E1, L2-E6, L1-E5]
2	1	300	Simple	[L2-E3, L1-E1, L1-E2]	[L2-E1, L2-E6, L1-E5]
2	3	100	Distance	[L1-E1, L2-E3, L2-E4]	[L2-E1, L1-E4, L1-E5]
2	3	100	Simple	[L1-E1, L2-E3, L2-E4]	[L1-E4, L2-E1, L1-E5]
2	3	300	Distance	[L1-E1, L2-E3, L2-E4]	[L2-E1, L1-E4, L1-E5]
2	3	300	Simple	[L1-E1, L2-E3, L2-E4]	[L1-E4, L2-E1, L1-E5]
4	1	100	Distance	[L2-E3, L1-E1, L1-E2]	[L2-E1, L2-E6, L1-E5]
4	1	100	Simple	[L2-E3, L1-E1, L1-E2]	[L2-E1, L2-E6, L1-E5]
4	1	300	Distance	[L2-E3, L1-E1, L1-E2]	[L2-E1, L2-E6, L1-E5]
4	1	300	Simple	[L2-E3, L1-E1, L1-E2]	[L2-E1, L2-E6, L1-E5]
4	3	100	Distance	[L1-E1, L2-E3, L2-E4]	[L2-E1, L1-E4, L1-E5]
4	3	100	Simple	[L1-E1, L2-E3, L2-E4]	[L1-E4, L2-E1, L1-E5]
4	3	300	Distance	[L1-E1, L2-E3, L2-E4]	[L2-E1, L1-E4, L1-E5]
4	3	300	Simple	[L1-E1, L2-E3, L2-E4]	[L1-E4, L2-E1, L1-E5]

Notes: Top contributors are the location-epoch decisions with the largest improvement contributions $I_j = -\bar{\phi}_j$. Lowest contributors are the location-epoch decisions with the smallest improvement contributions. The same realized demand and lead-time series are used across all parameter settings.

6. Summary

6.1. Conclusion

This paper proposes a Shapley-based decomposition framework for attributing inventory performance gaps to individual order decisions. We introduce a counterfactual contribution mechanism that allocates the realized performance difference between an observed policy and an optimized benchmark across decision epochs.

 doi:10.20944/preprints202603.2499.v2

The key innovation of the framework lies in treating decisions as contributors within a cooperative game defined over dynamic inventory evaluation. By averaging marginal effects across all possible correction contexts, the Shapley-based decomposition provides a straightforward insight into each decision's contribution to the gap between the desired and the observed evaluation results.

Compared to traditional knowledge-based reason analysis, sensitivity analysis, and variance-based decomposition, the proposed approach directly addresses a distinct diagnostic question: *which specific decisions explain the realized gap between two policies?* This focus on counterfactual gap attribution provides a decision-level perspective that complements existing inventory evaluation tools. This approach is also distinct from the traditional idea of Shapley value, with the former being more relevant to business applications and less computational burden.

From a managerial standpoint, the framework enhances transparency and accountability in inventory governance. It enables quantitative assessment of decision contributions, supports root-cause analysis of forecasting error or lead-time delay, and facilitates continuous policy improvement relative to optimized benchmarks. Because the method does not rely on differentiability or convexity, it can be integrated into modern analytics platforms built on simulation or data-driven optimization.

6.2. Future Research Direction

The decomposition framework can be extended beyond direct order quantities to analyze the underlying drivers of ordering decisions. In practice, order quantities are often determined by rules that depend on demand forecasts, safety stock calculations, and anticipated lead times. Moreover, realized arrivals may deviate from planned lead times due to logistical delays.

Let the order decision at epoch j be expressed as a function

$$x_j = f(\hat{D}_j, \hat{L}_j, \theta),$$

where $\hat{D}_j$ denotes demand forecasts, $\hat{L}_j$ denotes expected lead time, and θ represents policy parameters.

By redefining the "players" in the decomposition as components such as forecast inputs or realized lead-time deviations, the same counterfactual framework can be applied to attribute performance gaps to forecasting accuracy or delivery delays. For example, it is possible to replace realized demand forecasts with ideal or unbiased forecasts while holding other inputs fixed. It is also possible to replace realized lead times with planned lead times to measure the contribution of delay.

In this way, the method extends naturally from decision-level attribution to root-cause analysis, enabling managers to distinguish whether underperformance is primarily driven by forecasting error, execution delays, or discretionary ordering behavior.

Appendix A. Variance-Reduction Techniques for the Monte Carlo Shapley Estimator

Antithetic permutation sampling.

Antithetic sampling is a classical variance-reduction method in Monte Carlo simulation. The basic idea is to generate negatively correlated or complementary samples so that the average of paired samples has lower variance than the average of independently sampled values [19,20]. In the context of permutation-based Shapley estimation, this idea can be implemented by pairing each sampled permutation with its reverse order. Let

$$\pi = (\pi_1, \pi_2, \ldots, \pi_p)$$

be a sampled permutation of the decision epochs. Its antithetic counterpart is

$$\pi^{rev} = (\pi_p, \pi_{p-1}, \ldots, \pi_1).$$

 doi:10.20944/preprints202603.2499.v2

For each pair (π, π^{rev}), the marginal contribution of decision j is computed under both correction sequences and then averaged:

$$\Delta_j^{anti} = \frac{1}{2}\left(\Delta_j^{(\pi)} + \Delta_j^{(\pi^{rev})}\right).$$

This pairing is useful because a decision that appears early in π appears late in π^{rev}. Thus, the estimator samples both early-correction and late-correction contexts more evenly. Recent work on permutation sampling for Shapley value estimation also discusses antithetic sampling as a way to improve permutation-based Shapley approximations [23].

Stratified permutation sampling.

Stratified sampling reduces variance by ensuring that samples are drawn from different parts of the sampling space rather than relying purely on independent random draws [19,20]. In Shapley estimation, the marginal contribution of a decision epoch j may depend strongly on the number of other decisions that have already been corrected. Therefore, instead of sampling permutations completely at random, one may stratify by the position of decision j in the permutation, or equivalently by the coalition size $|A|$.

For example, the set of possible positions $\{1, \ldots, p\}$ can be divided into strata, and permutations can be sampled so that each decision epoch appears approximately equally often in early, middle, and late positions. This improves coverage of different coalition sizes and reduces the chance that the estimated attribution is driven by an unbalanced set of correction contexts. Sampling-based methods for Shapley value estimation have been studied in cooperative-game settings, and stratification is a natural extension for improving the statistical efficiency of such estimators [22,23].

References

1. Silver, Edward Allen, David F. Pyke, and Rein Peterson. Inventory management and production planning and scheduling. Vol. 3. New York: Wiley, 1998.
2. Fiacco, Anthony V. "Introduction to sensitivity and stability analysis in non linear programming." (1983).
3. Sobol, Ilya M. "Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates." Mathematics and computers in simulation 55.1-3 (2001): 271-280.
4. Saltelli, Andrea, et al. Global sensitivity analysis: the primer. John Wiley & Sons, 2008.
5. Shapley, Lloyd S. "A value for n-person games." (1953): 307-317.
6. Bartholdi III, John J., and Eda Kemahlioğlu-Ziya. "Using Shapley value to allocate savings in a supply chain." Supply chain optimization. Boston, MA: Springer US, 2005. 169-208.
7. Kemahlıoğlu-Ziya, Eda, and John J. Bartholdi III. "Centralizing inventory in supply chains by using Shapley value to allocate the profits." Manufacturing & Service Operations Management 13.2 (2011): 146-162.
8. Fiestras-Janeiro, M. Gloria, et al. "Cooperative game theory and inventory management." European Journal of Operational Research 210.3 (2011): 459-466.
9. Van den Heuvel, Wilco, Peter Borm, and Herbert Hamers. "Economic lot-sizing games." European Journal of Operational Research 176.2 (2007): 1117-1130.
10. Saltelli, Andrea, et al. "Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index." Computer physics communications 181.2 (2010): 259-270.
11. Lundberg, Scott M., and Su-In Lee. "A unified approach to interpreting model predictions." Advances in neural information processing systems 30 (2017).
12. Bertsimas, Dimitris, and John N. Tsitsiklis. Introduction to linear optimization. Vol. 6. Belmont, MA: Athena scientific, 1997.
13. Axsäter, Sven. Inventory control. Boston, MA: Springer US, 2006.
14. Graves, Stephen C. "A single-item inventory model for a nonstationary demand process." Manufacturing & Service Operations Management 1.1 (1999): 50-61.
15. Geevers, Kevin, Lotte Van Hezewijk, and Martijn RK Mes. "Multi-echelon inventory optimization using deep reinforcement learning." Central European Journal of Operations Research 32.3 (2024): 653-683.
16. Censor, Yair, et al. "On the effectiveness of projection methods for convex feasibility problems with linear inequality constraints." Computational Optimization and Applications 51.3 (2012): 1065-1088.

 doi:10.20944/preprints202603.2499.v2

17. Barratt, Shane, Guillermo Angeris, and Stephen Boyd. "Automatic repair of convex optimization problems." Optimization and Engineering 22.1 (2021): 247-259.
18. Harsha, Pavithra, et al. "Deep policy iteration with integer programming for inventory management." Manufacturing & Service Operations Management 27.2 (2025): 369-388.
19. Glasserman, Paul. *Monte Carlo Methods in Financial Engineering*. Springer, 2004.
20. Owen, Art B. *Monte Carlo Theory, Methods and Examples*. 2013.
21. Law, Averill M. *Simulation Modeling and Analysis*. 5th ed., McGraw-Hill Education, 2015.
22. Castro, Javier, Daniel Gómez, and Juan Tejada. "Polynomial calculation of the Shapley value based on sampling." *Computers & Operations Research* 36, no. 5 (2009): 1726–1730.
23. Mitchell, Rory, Joshua Cooper, Eibe Frank, and Geoffrey Holmes. "Sampling permutations for Shapley value estimation." *Journal of Machine Learning Research* 23, no. 43 (2022): 1–46.