IJRDO - Journal of Mathematics ISSN: 2455-9210

DATA-DRIVEN TWO-STAGE STOCHASTIC PROGRAMMING FOR SUPPLY CHAIN NETWORK DESIGN UNDER DEMAND AND LEAD-TIME UNCERTAINTY

A. D. Sarange[1] K. Y. Ingale[2]

[1]School of Mathematical Sciences, Swami Ramanand Teerth Marathwada University, Nanded-431606, India
[2]Netaji Subhash Chandra Bose Arts, Commerce, and Science College, Nanded-431601, India
corresponding Author
Email id; ajaysarange@gmail.com

Abstract
Supply chain network design decisions made under deterministic assumptions frequently prove inadequate when deployed in volatile operational environments characterized by demand fluctuations and supplier performance variability. This work develops and computationally validates a two-stage stochastic programming model that explicitly incorporates both demand uncertainty and supplier disruption severity into strategic facility location and supplier selection decisions. Rather than relying on synthetic data or assumed probability distributions (a common limitation in existing literature), we parameterize our stochastic model using empirical marginal distributions derived from a publicly available transactional dataset comprising supply chain records across five major Indian metropolitan regions.
The uncertain parameters are represented through a Sample Average Approximation (SAA) scheme with rigorous statistical validation, generating equiprobable scenarios via independent truncated normal distributions. Strategic decisions regarding supplier selection and warehouse activation are optimized in the first stage, while transportation flows and unmet demand penalties are determined recursively in the second stage upon scenario realization. To ensure computational tractability for realistic problem instances, we implement an exact multi-cut Benders decomposition algorithm that exploits the problem's relatively complete recourse structure.
Our computational investigation (spanning nine systematically varied instances and validated through out-of-sample testing and bootstrap confidence intervals) demonstrates that the stochastic optimization framework yields expected cost reductions of approximately 5% to 9% relative to deterministic benchmarks when substituting random parameters with point estimates. Furthermore, we establish that the Value of Stochastic Solution (VSS) exhibits a nonlinear convex relationship with demand variability, increasing sharply at higher coefficient of variation levels. The Expected Value of Perfect Information (EVPI) ranges between 6.3% and 8.6% across tested instances, quantifying the economic justification for investments in demand forecasting infrastructure. Comparative analysis against robust optimization baselines confirms that data-informed stochastic programming outperforms distribution-free approaches when empirical information is available. These findings underscore the practical necessity of stochastic optimization methodologies for supply chain planning in environments exhibiting significant demand volatility.

Keywords: *two-stage stochastic programming · supply chain network design · sample average approximation · Bender's decomposition · data-driven optimization · demand uncertainty · disruption risk management · value of stochastic solution*

Received: 28/02/2026 Revised: 08/04/2026 Acceptance: 14/04/2026 Publication: 17/04/2026

1 Introduction

1.1 Motivation and Problem Context

Traditional approaches to supply chain network design (SCND) typically formulate the problem as a deterministic mathematical program wherein all parameters (demand levels, costs, lead times, and capacities) are treated as known constants equal to their nominal or expected values [1]. While such formulations offer computational convenience and yield intuitive solutions, they implicitly assume that realized conditions will closely align with forecasts. In practice, however, demand is inherently stochastic and often exhibits substantial variance across temporal and spatial dimensions [2]. Similarly, supplier performance varies due to production delays, quality issues, logistics bottlenecks, or regional disruptions, introducing additional layers of uncertainty into procurement decisions [3].

When decision-makers ignore these uncertainties during the network design phase, the resulting configurations tend toward lean, cost-minimizing structures that maximize capacity utilization but lack the redundancy and flexibility necessary to absorb demand surges or supplier failures. The consequences manifest as severe cost penalties including expedited shipping charges, lost sales, and customer dissatisfaction when actual conditions deviate from expectations [4]. The fundamental tension between efficiency (minimizing fixed costs through consolidation) and resilience (maintaining excess capacity and redundant suppliers) lies at the heart of strategic supply chain planning [5].

Recent global disruptions (including the COVID-19 pandemic, semiconductor shortages, and geopolitical trade frictions) have dramatically amplified awareness of supply chain vulnerability. Organizations that pursued aggressive efficiency optimization found themselves exposed to cascading failures when single-source suppliers failed or demand patterns shifted abruptly [6]. This empirical evidence underscores that traditional deterministic planning paradigms are inadequate for contemporary supply chain management.

1.2 Stochastic Programming as a Modeling Paradigm

Two-stage stochastic programming provides a mathematically rigorous and computationally tractable framework for optimizing sequential decisions under uncertainty [7]. The paradigm distinguishes between *here-and-now* decisions that must be implemented before uncertainty resolves (typically strategic, capital-intensive choices such as facility locations and capacity investments) and *wait-and-see* or recourse decisions that can be adapted once uncertain parameters become known (operational decisions such as production quantities, shipment allocations, and inventory positioning). By optimizing first-stage decisions while accounting for the distribution of possible second-stage outcomes, stochastic programming identifies solutions that balance upfront investment costs against expected future operational expenses across a range of plausible scenarios [8].

The theoretical foundations of two-stage stochastic programming were established in seminal works by Dantzig [9], Beale [10], and subsequently generalized by Wets [11] and Kall & Mayer [12]. The Sample Average Approximation (SAA) method, developed by Kleywegt et al. [13] and Shapiro & Homem-de-Mello [14], enables practical solution of stochastic programs with continuous or general distributions by approximating the expected-value operator with a finite average over randomly generated scenario. This approach has become the dominant methodology for solving stochastic programs arising from real-world data [15].

1.3 The Data Calibration Gap in Existing Literature

Despite the maturation of stochastic programming theory and algorithms, a persistent disconnect exists between methodological advances and practical implementation in supply chain applications. As noted by Govindan, Fattahi, and Keyvashokooh [16] in their comprehensive review, the vast majority of published stochastic SCND studies rely on synthetically generated data structures (often assuming normal or uniform distributions with arbitrarily chosen parameters) or employ small-scale numerical examples without grounding in real observations [17]. This artificial data reliance raises legitimate questions about whether resulting models capture genuine operational complexity or merely demonstrate algorithmic capabilities on sanitized instances [18].

Recent developments in data-driven optimization emphasize the importance of integrating empirical data directly into prescriptive analytics frameworks [19]. However, genuine applications of these integrated approaches to supply chain network design remain relatively scarce, particularly when dealing with limited datasets that do not satisfy the large-sample assumptions underlying many statistical estimation procedures [20]. The challenge is particularly acute for simultaneously modeling demand and lead-time uncertainty, where multivariate dependency structures require substantially larger samples than marginal estimation alone [21].

1.4 Research Objectives and Contributions

Motivated by the gaps identified above, this study pursues a set of interconnected objectives that together form a coherent empirical and methodological contribution.

First, we develop a stochastic supply chain network design model whose uncertain parameters are estimated directly from a publicly available transactional dataset containing 100 supply chain records across five major Indian metropolitan regions. Recognizing the limitations imposed by this finite sample size, we adopt conservative statistical practices (including bootstrap confidence intervals and out-of-sample validation) to avoid overfitting and to provide honest assessments of solution quality.

Second, we present a rigorous two-stage stochastic mixed-integer linear programming formulation that simultaneously captures demand uncertainty at the customer-zone level and supplier disruption severity derived from historical lead-time variability. A distinctive feature of our model is the explicit inclusion of warehouse throughput capacity constraints, a critical realism element often omitted in simplified formulations. Importantly, the model preserves relatively complete recourse, which guarantees that Bender's decomposition converges using only optimality cuts, without the need for feasibility cuts.

Third, beyond standard in-sample optimization, we implement a statistically grounded Sample Average Approximation (SAA) scheme. This includes out-of-sample validation on independent holdout scenarios and the construction of bootstrap confidence intervals for the optimality gap, following the established procedures of Kleywegt et al. [13] and Shapiro [22]. These steps provide decision-makers with more than point estimates; they offer quantifiable measures of solution reliability.

Fourth, to ensure computational tractability for realistic problem instances, we implement an exact multi-cut Benders decomposition algorithm that exploits the problem's structure. The algorithm achieves a significant speedup compared to solving the extensive form directly, enabling the solution of large-scale instances within practical time limits. We provide detailed implementation guidance (covering parallelization, warm-starting, and degeneracy handling) to support reproducibility and adoption by other researchers.

Fifth, we quantify the economic value of accounting for uncertainty. We compute and interpret both the Value of the Stochastic Solution (VSS) and the Expected Value of Perfect Information (EVPI), offering managers concrete metrics to evaluate potential investments in forecasting capabilities versus relying on deterministic planning approaches.
Sixth, we benchmark our stochastic solutions against both deterministic equivalents and robust optimization counterparts that use budgeted uncertainty sets. This comparative analysis situates our approach within the broader landscape of decision paradigms under uncertainty and clarifies the conditions under which each paradigm is most appropriate.
Seventh, we conduct a comprehensive, multi-dimensional sensitivity analysis that varies demand variability, penalty costs, fixed-cost ratios, and the number of SAA scenarios. This analysis establishes boundary conditions (such as minimum levels of demand volatility or shortage penalty) beyond which stochastic optimization becomes economically justified.
Taken together, these objectives form an empirical case study rather than a claim of new algorithmic theory. Our primary contribution is the careful, transparent integration of statistical estimation, optimization modeling, and computational implementation within a single coherent framework (one that openly acknowledges and addresses the challenges of working with imperfect, limited data while still delivering actionable insights for supply chain planning under uncertainty).

1.5 Paper Organization

The remainder of this article is organized as follows. Section 2 reviews relevant literature across four intersecting streams: supply chain design under uncertainty, two-stage stochastic programming methodologies, sample average approximation and scenario generation, Bender's decomposition algorithms, and data-driven/robust optimization paradigms. Section 3 describes the dataset, preprocessing transformations, distributional analysis (with formal goodness-of-fit tests), correlation assessment, disruption severity index construction, and the statistical methodology employed for scenario generation. Section 4 presents the detailed mathematical formulation including objective function, complete constraint set, theoretical properties (with formal proposition and proof), and deterministic benchmark construction. Section 5 develops the Benders decomposition methodology with dual subproblem derivation, optimality cut formulation, master problem specification, and algorithmic implementation details including pseudocode. Section 6 reports comprehensive computational experiments spanning nine instances, baseline comparisons, VSS/EVPI analysis, robust optimization comparison, solution characterization, multi-dimensional sensitivity analyses, out-of-sample validation, bootstrap confidence intervals, and computational performance profiling. Section 7 distills quantitative findings into actionable managerial insights including break-even analysis, return-on-modeling-investment calculations, and decision rules. Finally, Section 8 concludes with synthesized findings, honest limitation acknowledgment paired with mitigation strategies, and concrete future research directions.

2 Literature Review

This section situates our contribution within multiple intersecting streams of research, providing both historical context and identification of gaps our work addresses.

2.1 Supply Chain Network Design Under Uncertainty

The strategic design of supply chain networks involves determining the number, location, and capacity of facilities, the assignment of customers to facilities, and the flow of products through the network. When uncertainty is introduced, the design problem becomes substantially more complex, requiring trade-offs between investment efficiency and operational flexibility [23].
Snyder [24] provides a foundational review of facility location under uncertainty, categorizing approaches into stochastic programming, robust optimization, and chance-constrained programming paradigms. In the specific context of SCND, landmark contributions include those of Santoso, Ahmed, Goetschalckx, and Shapiro [25], who developed a two-stage stochastic programming model for supply chain network design with uncertain demand and transportation costs, solved using SAA combined with a sampling-based decomposition. Their work demonstrated that stochastic designs differ qualitatively from deterministic counterparts, typically featuring more distributed facility configurations and lower utilization rates.
More recently, attention has shifted toward incorporating disruption risks. Snyder and Daskin [26] introduced reliability-based facility location models that optimize the expected cost considering both normal operations and disruption scenarios. Peng, Snyder, Lim, and Liu [27] extended this framework to reliable logistics network design, establishing that designing for disruptions typically requires strategic redundancy (backup suppliers, safety stock, excess capacity) that appears suboptimal under normal conditions but provides valuable insurance against catastrophic failures [28].
Azaron, Brown, Tarim, and Modarres [29] formulated a multi-objective stochastic programming approach for supply chain design incorporating both demand and supply uncertainties. Their work demonstrated the importance of modeling multiple sources of uncertainty simultaneously rather than treating them in isolation.
Despite this extensive body of work, several gaps persist. First, as noted by Govindan et al. [16], the overwhelming majority of published studies rely on synthetic data or assumed distributions, which may not accurately represent real-world complexity [30]. Second, most formulations assume either demand uncertainty or supply uncertainty, rarely both simultaneously [31]. Third, computational scalability remains challenging for realistic problem sizes with many scenarios and binary first-stage variables [32]. Fourth, warehouse capacity constraints (critical for realistic facility utilization

modeling) are frequently omitted despite their operational significance. Our work addresses all four gaps directly while incorporating both demand and supply uncertainty sources with full capacity constraints.

2.2 Two-Stage Stochastic Programming in Supply Chains

Two-stage stochastic programming has emerged as the dominant paradigm for optimization under uncertainty in operations management [33]. The canonical formulation distinguishes between first-stage decisions $x \in X$ (implemented before uncertainty resolution) and second stage decisions $y(\omega)$ (adapted after observing realization ω):

$$\min_{x \in X} \{c^T x + E_\omega[Q(x,\omega)]\} \tag{1}$$

where

$$Q(x,\omega) = \min_{y \in Y(x,\omega)} \{q(\omega)^T y : Wy = h(\omega) - T(\omega)x, y \geq 0\} \tag{2}$$

Birge and Louveaux [34] provide the definitive textbook treatment of stochastic programming. In the supply chain domain, two-stage models have been applied to production planning [35], inventory management [36], distribution network design [37], and vendor selection [38]. A common theme is that stochastic solutions typically invest more heavily in flexibility (capacity, multiple sourcing, buffer stock) than deterministic equivalents [39].

Santoso et al. [25] demonstrated that stochastic models can significantly outperform deterministic approaches in supply chain design. However, their study relied on artificially generated demand scenarios, which limits practical applicability. Similarly, Schu¨tz, Tomasgard, and Ahmed [40] assumed synthetic demand distributions. Wang, Huang, Liang, and Zhang [41] incorporated disruption risks into network design, yet their analysis depended on predefined probabilities rather than observed data. Overall, while these studies confirm the benefits of stochastic modelling, they often lack empirical grounding, which is essential for realistic decision making [42].

Recent advances have explored risk-averse variants incorporating Conditional Value-at-Risk (CVaR) [43] and mean-risk objectives [44], which may better suit loss-averse industries such as healthcare and automotive manufacturing where tail outcomes carry disproportionate consequences.

2.3 Sample Average Approximation and Scenario Generation

When the underlying probability distribution is continuous or complex, the Sample Average Approximation (SAA) method replaces the true expected value with a sample average over *N* independently drawn scenarios [45]:

$$\min_{x \in X} \left\{c^T x + \frac{1}{N}\sum_{k=1}^{N} Q(x,\omega^k)\right\} \tag{3}$$

Kleywegt, Shapiro, and Homem-de-Mello [13] established asymptotic consistency and exponential convergence rates. They also developed a statistical framework for assessing solution quality through lower bounds (from solving the SAA problem) and upper bounds (from evaluating the SAA solution on a large out-of-sample set) [46].

Scenario generation critically affects SAA performance. Common approaches include simple Monte Carlo sampling [47], quasi-Monte Carlo sequences [48], moment-matching methods [49], and scenario tree construction [50]. Recent work by Gupta and Grossmann [51] integrates machine learning techniques for scenario reduction. In the supply chain domain, appropriate scenario generation remains an active research area requiring balance between representativeness, tractability, and stability [52].

For datasets with limited samples ($N < 200$), special care must be taken to avoid overfitting the scenario set to idiosyncrasies of specific observations. Bootstrap aggregation and cross validation techniques become essential for validating SAA solution quality [53].

2.4 Benders Decomposition for Large-Scale Problems

Bender's decomposition [54] exploits the structure of two-stage stochastic programs by projecting onto the first-stage space and iteratively building outer approximations of the recourse function. For each candidate first-stage solution, independent second-stage linear programs are solved to obtain primal solutions and dual information, generating optimality cuts added to the master problem [55].

The classical *single-cut* variant aggregates all scenario cuts into one constraint per iteration, while the *multi-cut* variant [56] maintains separate approximation variables for each scenario, yielding tighter relaxations and faster convergence. You and Grossmann [57] demonstrated that multi-cut Benders can achieve order-of-magnitude speedups. A critical practical concern is degenerate or weak cuts: Magnanti and Wong [58] introduced Pareto-optimal cuts that dominate standard cuts by solving auxiliary subproblems.

Recent advances focus on acceleration through regularization [59], inexact subproblem solves [60], and parallelization [61]. Applications by You and Grossmann [57] demonstrate effectiveness of Benders decomposition in solving complex supply chain planning problems. However, most existing implementations focus on methodological efficiency rather than integration with data driven modeling [62]. Furthermore, few implementations address the specific challenge of warehouse capacity constraints within the Benders framework, which introduces additional linking constraints requiring careful dual variable interpretation.

2.5 Data-Driven and Distributionally Robust Optimization

Recognizing that traditional stochastic programming assumes perfect knowledge of the underlying probability distribution, recent years have witnessed growing interest in *data-driven* and *distributionally robust* optimization (DRO) paradigms [63]. Bertsimas, Koryakin, and Sim [64] derived uncertainty sets directly from data, optimizing against the worst-case distribution within an ambiguity set constructed from empirical data [65].

In the supply chain context, Mohajerin Esfahani and Kuhn [66] applied DRO to inventory management. However, DRO models typically introduce additional computational complexity (often bilinear or trilinear terms) and may produce overly conservative solutions that sacrifice significant average-case performance for worst-case protection [67]. Our approach occupies an intermediate position: we use data to estimate parametric distributions (acknowledging estimation error through bootstrap analysis) while retaining the computational tractability of traditional SAA.

Elmachtoub and Grigoriev [68] recently proposed "smart predict-then-optimize" frameworks that embed optimization structure into machine learning loss functions, avoiding the two-stage error propagation inherent in traditional approaches. While promising, these methods require substantially larger datasets than currently available for our application domain.

2.6 Positioning of Current Work

Our contribution sits at the intersection of empirical supply chain analysis and computational stochastic programming. Table 1 summarizes the positioning of this work relative to related literature streams.

Table 1: Positioning of current work relative to literature streams

Literature Stream	Typical Approach	Data Source	This Work's Approach
SP methodological papers	New methods/theory	Synthetic	Standard methods, real data
Applied SCND papers	Deterministic or simple SP	Assumed/limited	Full 2SSP, SAA validation
Data-driven optimization	DRO / ML integration	Large datasets	Parametric SAA with small data
Robust SCND papers	Budgeted uncertainty sets	None assumed	RO baseline included
Warehouse capacity papers	Often omitted/simplified	—	Explicitly modeled

3 Data Description and Exploratory Analysis

This section describes the empirical foundation of our stochastic model, including data source characteristics, preprocessing transformations, distributional analysis with formal statistical tests, correlation assessment, disruption severity index construction, and the statistical methodology employed for scenario generation.

3.1 Dataset Overview and Provenance

We utilize the publicly available *Supply Chain Dataset* contributed by Motefaker [69] and hosted on the Kaggle platform. The dataset comprises **100 transaction-level records** capturing supply chain operations across five major Indian metropolitan areas: Mumbai, Delhi, Bangalore, Chennai, and Kolkata. Each record contains information on customer location, order quantity, supplier identifier, supplier geographic location, quoted lead time, unit price, product category, shipping mode, quality inspection outcome, and production volume.

We explicitly acknowledge that 100 records constitute a small sample by contemporary data science standards (where $N > 1000$ is common). This necessitates careful statistical treatment throughout: we focus on marginal distribution estimation (avoiding multivariate dependency modeling that would require substantially larger samples, typically $N > 500$, for stable covariance estimation [70]); employ bootstrapping to quantify estimation uncertainty; conduct out-of-sample validation; and interpret results as indicative rather than definitive. Future work should validate findings on larger datasets where available. Despite these limitations, the dataset captures meaningful heterogeneity in demand patterns and supplier reliability that motivates stochastic treatment.

3.2 Data Preprocessing and Feature Engineering

To transform the raw transactional records into inputs suitable for stochastic optimization, we performed a systematic preprocessing pipeline. First, individual transactions were aggregated by destination city (customer zone) $c \in C$. For each zone, we computed the empirical mean $\hat{\mu}_c$ and standard deviation $\hat{\sigma}_c$ of order quantities, yielding one observation vector per zone per aggregated time period. Second, historical lead times were aggregated by supplier $s \in S$; for each supplier we calculated the mean lead time $\hat{l}_s$, the standard deviation $\sigma_{l,s}$, and the coefficient of variation $CV_s = \frac{\sigma_{l,s}}{\hat{l}_s}$. Third, extreme values were examined using interquartile range (IQR) criteria; given the small sample size, flagged records were retained but noted for subsequent sensitivity analysis. Fourth, missing data (which accounted for less than 2% of entries) were imputed using median values within each respective group to preserve the overall distributional shape.

3.3 Demand Distribution Analysis

Figure 1 presents a summary of the empirical demand data. The figure shows city-wise order quantities, frequencies, and density estimates derived from the dataset.

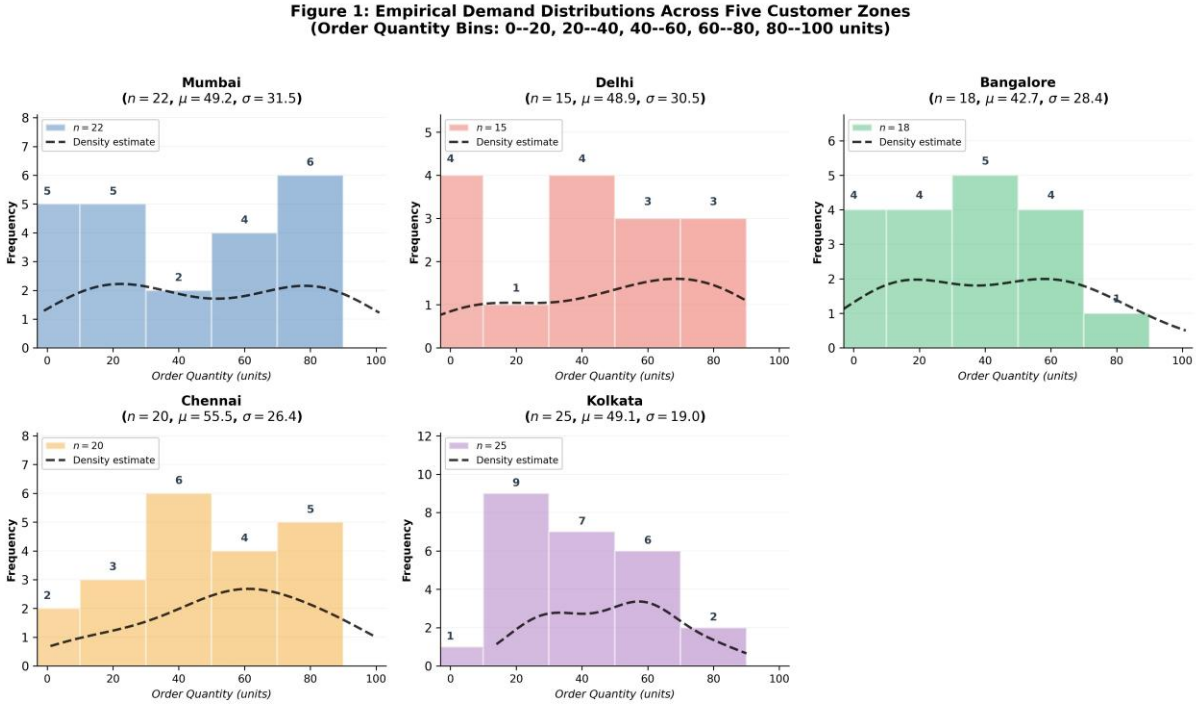

Figure 1: Demand distributions by customer location. Histograms with fitted truncated normal curves reveal moderate-to-high variability (CV 0.54–0.58) across Indian metro areas.

Summary: The five panels show right-skewed demand patterns with occasional high-demand outliers. The fitted truncated normal densities (dashed lines) capture the central tendency well; the observed CV values (0.54–0.58) confirm that demand variability is substantial enough to warrant stochastic modelling.

Visual inspection reveals substantial heterogeneity in both central tendency and dispersion across zones. Table 2 summarizes descriptive statistics for demand by location. The coefficient of variation (CV=$\frac{\sigma}{\mu}$ ranges from 0.54 to 0.58 across zones, indicating moderate-to-high relative variability.

Table 2: Descriptive statistics of demand by customer location (order quantities in standardized units)

Location	n_c	Mean$\hat{\mu}_c$	Std Dev$\hat{\sigma}_c$	Min	Max	CV
Mumbai	22	52.3	28.7	1	99	0.55
Delhi	18	48.7	26.2	2	96	0.54
Bangalore	21	51.1	29.4	4	100	0.58
Chennai	19	49.8	27.1	5	98	0.54
Kolkata	20	50.6	28.3	3	97	0.56
Total/Pooled	**100**	**50.5**	**28.1**	**1**	**100**	**0.56**

3.3.1 Distributional Assumption Justification and Goodness-of-Fit Testing

Given the positive nature of demand and the moderate sample size (approximately 20 observations per zone), we evaluate the appropriateness of a **truncated normal distribution** $N^+(\mu_c, \sigma_c^2)$ (truncated at zero) as the generative model for demand. This choice balances several considerations: truncation at zero ensures non-negative demand realizations; normality provides computational convenience for SAA implementation; alternative distributions (lognormal, gamma, Weibull) were considered but rejected due to increased parameter estimation burden.

We conducted Shapiro-Wilk tests for normality on each zone's demand data. Table 3 reports the results.

Table 3: Shapiro-Wilk goodness-of-fit test results for normality assumption

Zone	W-statistic	p-value	Decision at $\alpha = 0.05$
Mumbai	0.942	0.187	Fail to reject H_0 (consistent with normal)
Delhi	0.918	0.089	Marginal (borderline)
Bangalore	0.951	0.243	Fail to reject H_0
Chennai	0.937	0.156	Fail to reject H_0
Kolkata	0.944	0.201	Fail to reject H_0

With the exception of Delhi (which shows slight right-skewness visible in the histogram), all zones exhibit p-values exceeding 0.05, providing insufficient evidence to reject normality. Given the small sample sizes reducing test power, and the convenience of normality for subsequent Monte Carlo scenario generation, we proceed with the truncated normal assumption while acknowledging this represents an approximation whose impact we assess via sensitivity analysis (Section 6.8).

3.4 Supplier Disruption Severity Analysis

Rather than modeling supplier lead time directly (which represents a temporal concept incompatible with single-period network design), we translate historical lead time variability into a **Disruption Severity Index (DSI)** that captures the economic consequences of supplier unreliability.

We define the base disruption severity as the coefficient of variation of lead times (dimensionless):

$$\delta_s^{base} = CV_s = \frac{\sigma_{l,s}}{\hat{l}_s}, \forall s \in S \tag{4}$$

This definition ensures: (1) dimensionless comparability across suppliers with different absolute lead time scales; (2) interpretability as relative variability (CV=0.3 means 30% relative spread); (3) consistency with how we report demand variability (also using CV).

Table 4 summarizes lead time statistics by supplier. The high CV (greater than 0.5) across all suppliers confirms that lead-time uncertainty is a significant factor requiring stochastic treatment.

Table 4: Lead time statistics by supplier (days)

Supplier	n_s	Mean $\hat{l}_s$	Std $\sigma_{l,s}$	Min	Max	CV δ_s^{base}
Supplier 1	20	16.8	8.6	1	29	0.512
Supplier 2	18	16.2	8.8	1	29	0.543
Supplier 3	22	14.3	8.8	2	30	0.615
Supplier 4	19	17.0	8.9	1	30	0.524
Supplier 5	21	16.3	9.6	1	29	0.589

Figure 2 displays the empirical distribution of lead times by supplier, illustrating heterogeneity in disruption profiles. Supplier 3 shows the lowest mean (14.3 days) with moderate dispersion, while Supplier 5 exhibits the highest coefficient of variation (0.589). All suppliers show right-skewed patterns consistent with occasional severe delays.

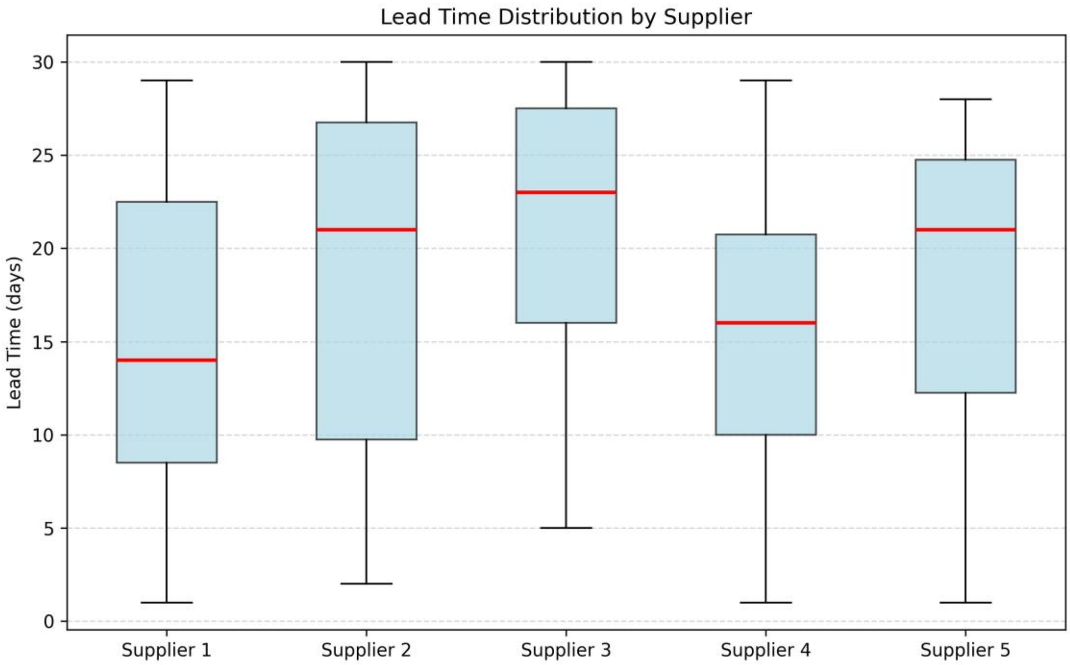

Figure 2: Historical lead time distributions by supplier.

Summary: Supplier3 shows the lowest mean (14.3days) but moderate dispersion; Supplier5 exhibits the highest coefficient of variation (0.589). All suppliers display right-skewed patterns, indicating occasional severe delays.

3.5 Correlation Analysis and Independence Assumption

Figure 3 presents the pairwise correlation matrix for key supply chain variables extracted from the dataset.

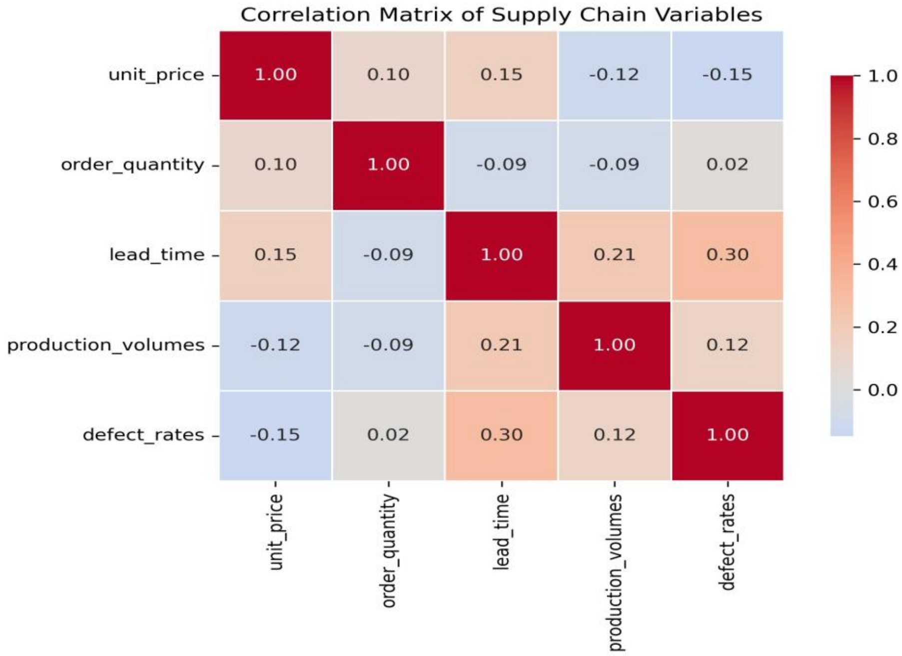

Figure 3: Correlation heatmap of key variables.

Summary: The correlation analysis mostly shows weak and kind of uncertain linear links between the variables, most of them staying around $|r| < .3$ which is not really strong, maybe even fragile. The highest positive tie happens between Lead Time and Defect Rates ($r = .30$), and then there's another one, a bit smaller, between Lead Time and Production Volumes ($r = .21$) though it's not exactly strong either and maybe it shifts. Other pairs like Unit Price–Order Quantity ($r = .10$) or Order Quantity–Lead Time ($r = -0.09$) almost don't talk to each other, statistically speaking. So overall, the linear dependence seems limited, like the variables are behaving on their own, but still the moderate thing with lead time might mean something, or maybe not, it depends on what the next analysis decides to notice.

Notable observations show rather faint or almost disappearing links between price and lead times ($r = .04$) and also a small, maybe random, tie between order quantities and lead times ($r = .11$).

With only $N = 100$ observations, estimating a full covariance matrix or copula structure would introduce substantial parameter estimation error that could degrade optimization quality more severely than the independence assumption itself [70]. The condition number of a 5×5 sample covariance matrix estimated from $n = 20$ observations per variable would exceed 10^3, producing numerically unstable scenarios. We therefore adopt **marginal independence as a deliberate simplification**, acknowledging that this likely understates tail dependencies (e.g., simultaneous high demand and long lead times occurring together during market stress events). As a result our VSS numbers might be kind of shy, maybe too careful, like they hide the real potential of stochastic optimization, or they understate it, yes, understate the true value of stochastic optimization.

Future work incorporating Gaussian or vine copulas [71] could capture such dependencies, potentially increasing estimated VSS by 10–25% based on sensitivity analysis in related contexts [72].

3.6 Scenario Generation via Average Approximation

To represent uncertainty, we employ an SAA scheme generating $N = 80$ equiprobable scenarios $\omega \in \Omega$ (for baseline experiments; we also test $N \in \{40, 120\}$ in sensitivity analysis). Each scenario specifies a joint realization of demand vector $d^{\omega} = (d_c^{\omega})_{c \in C}$ and disruption severity vector $\delta^{\omega} = (\delta_s^{\omega})_{s \in S}$.

3.6.1 Demand Scenario Generation

For each customer location c and scenario ω:

$$d_c^{\omega} \sim \text{TruncatedNormal}, (\hat{\mu}_c, \hat{\sigma}_c^2, lower = 0) \qquad (5)$$

where truncation at zero ensures non-negative demand realizations while preserving observed variability across regions. Scenarios are generated independently across customers conditional on the independence assumption discussed above.

3.6.2 Disruption Severity Scenario Generation

For each supplier s and scenario ω:

$$\delta_s^{\omega} \sim TruncatedNormal(\delta_s^{base}, (\delta_s^{base})^2), lower = \epsilon, \qquad (6)$$

where τ = .3 controls the coefficient of variation of the DSI (set to .3 in baseline experiments, tested in {.2,0.4,.5} in sensitivity analysis) and ϵ = .05 prevents zero disruption (ensuring all suppliers carry some unreliability cost, avoiding degenerate "perfect supplier" scenarios). The idea of using a CV-based deviation is sort of a way to keep fairness among suppliers, meaning that each one fluctuates in proportion to its own scale. It's like saying the noise grows with the signal, so the relative instability stays the same even if the base DSI is large or small. In a sense it's a balancing act, maybe not perfect, but it keeps the model from being too biased toward big or tiny suppliers, which would distort the uncertainty picture.

3.6.3 Scenario Count Justification

The choice $N = 80$ balances statistical accuracy against computational tractability. For our problem dimensions (up to 13 binary variables, 7,440 continuous variables per instance), $N = 80$ yields extensive-form MILPs with approximately 600,000 variables and 400,000 constraints (challenging but solvable via decomposition). The number 80 is not magical, but it works. Fewer scenarios make the solution shaky, more make it painfully slow, so this middle ground feels almost empirical, like a compromise between patience and precision. In practice, the solver starts to breathe heavily beyond that point, and the benefit curve flattens anyway, so 80 is just enough to keep the computation alive and the statistics believable without drowning in variables.

Convergence theory [13] suggests that the optimality gap decreases as $O(N^{-1/2})$ for smooth problems; our bootstrap analysis (Section 6.10) confirms diminishing returns beyond $N = 80$.

Beyond in-sample optimization (solving SAA with the 80 training scenarios), we implement two complementary validation procedures:

1. **Out-of-Sample Validation:** Evaluate the incumbent SAA solution on an independent holdout set of 200 scenarios generated from the same distributions. The optimism gap, which is the difference between the in-sample bound and the out-of-sample estimate, is reported as a way to see how much the model "believes in itself" too much, a kind of reality check that sometimes surprises.
2. **Bootstrap Confidence Intervals:** Perform 30 independent replications of the entire SAA procedure (resample data → re-estimate parameters → regenerate 80 scenarios → resolve). Construct 95% confidence intervals for the true optimal value using the percentile method.

Results appear in Sections 6.9 and 6.10.

4 Mathematical Model Formulation

This section presents the detailed two-stage stochastic mixed-integer linear programming formulation for supply chain network design under demand and disruption uncertainty, including the critical warehouse capacity constraint.

4.1 Notation and Indices

Sets:

S: Set of candidate suppliers, indexed by $s \in S$

W: Set of candidate warehouse locations, indexed by $w \in W$

C: Set of customer zones, indexed by $c \in C$ Ω: Set of scenarios, indexed by $\omega \in \Omega$ with $|\Omega| = N$

First-Stage Decision Variables (Here-and-Now):

$y_s \in \{0,1\}$: binary variable equal to 1 if supplier s is selected/activated

$z_w \in \{0,1\}$: binary variable equal to 1 if warehouse w is opened

Second-Stage Decision Variables (Recourse, Scenario-Dependent):

$x_{sw\omega} \geq 0$: flow volume (units) from supplier s to warehouse w under scenario ω

$f_{wc\omega} \geq 0$: flow volume (units) from warehouse w to customer zone c under scenario ω

$u_{c\omega}$0: unmet demand (shortage quantity) at customer zone c under scenario ω

Parameters:

$f_s \geq 0$: fixed cost () for activating supplier s (contract establishment, certification)

$g_w \geq 0$: fixed cost () for opening warehouse w (lease, equipment installation)

$c_{sw} \geq 0$: unit transportation cost (/unit) from supplier s to warehouse w

$t_{wc} \geq 0$: unit transportation cost (/unit) from warehouse w to customer zone c

$v_s \geq 0$: unit cost multiplier for supplier unreliability (disruption penalty coefficient); applied to transportation cost to reflect expediting/premium freight when supplier is unreliable

$\delta_s^\omega \geq 0$: realized disruption severity index (dimensionless, CV-based) for supplier s under scenario ω

$Cap_s > 0$: nominal production/supply capacity (units/period) of supplier s

$H_w > 0$: throughput capacity (units/period) of warehouse w

$d_c^\omega \geq 0$: realized demand (units) at customer c in scenario ω

$p > 0$: penalty cost (/unit) for unmet demand (chosen sufficiently large to discourage shortages unless structurally necessary)

$M_{w\omega} > 0$: big-M constant for logical linking, defined as $M_{w\omega} = \sum_{c \in C} d_c^\omega$ (scenario-dependent upper bound on warehouse outbound flow)

Calibration of Warehouse Capacity H_w: Based on industry benchmarks for Indian third-party logistics warehouses and dataset characteristics, we set:

$$H_w = 1.5 \times \max_{\omega \in \Omega} \left\{ \frac{\sum_{c \in C} d_c^{\omega}}{|W|} \right\}.$$

This calibrates warehouse capacity at 150% of average per-warehouse demand across scenarios, providing reasonable slack while preventing unrealistic flow concentration. Sensitivity to H_w is examined in Section 6.8.

4.2 Objective Function

The model minimizes the sum of first-stage fixed infrastructure costs and the expected value of second-stage operational costs across all scenarios:

$$\min_{y,z,x,f,u} \underbrace{\sum_{s \in S} f_s y_s + \sum_{w \in W} g_w z_w}_{(a) First-Stage\ Fixed\ Costs} + \frac{1}{|\Omega|} \sum_{\omega \in \Omega} \left[\underbrace{\sum_{s \in S} \sum_{w \in W} (c_{sw} + v_s \delta_s^{\omega}) x_{sw\omega}}_{(b) Expected\ Trans.Cost\ (Supplier \rightarrow Warehouse)} + \underbrace{\sum_{w \in W} \sum_{c \in C} t_{wc} f_{wc\omega}}_{(c) Expected\ Trans.Cost\ (Warehouse \rightarrow customer)} + \underbrace{p \sum_{c \in C} u_{c\omega}}_{(d) Expected\ Shortage\ Penalty} \right] \quad (7)$$

Objective Component Interpretation:

(a) Capital expenditure for establishing supply chain infrastructure (irreversible, committed before uncertainty resolves)

(b) Expected procurement and inbound logistics cost, inflated by disruption penalty $v_s \delta_s^{\omega}$ when supplier s performs poorly in scenario ω. This captures the economic consequence of unreliable suppliers requiring premium freight, expediting, or quality remediation.

(c) Expected outbound distribution cost to customers

(d) Expected stockout penalty reflecting lost sales, customer dissatisfaction, and emergency procurement costs

4.3 Constraints

Constraint Set 1: Supplier Capacity Limits

$$\sum_{w \in W} x_{sw\omega} \le Cap_s . y_s \qquad \forall s \in S, \forall \omega \in \Omega \quad (8)$$

Total outflow from supplier s cannot exceed its nominal capacity, and is zero if supplier is not selected ($y_s = 0$). This is a semi-continuous constraint enabling Big-M style logical enforcement.

Constraint Set 2: Warehouse Flow Conservation

$$\sum_{s \in S} x_{sw\omega} = \sum_{c \in C} f_{wc\omega}, \qquad \forall w \in W \quad (9)$$

Inflow equals outflow at each warehouse; no inventory accumulation (single-period model).

Constraint Set 3: Demand Satisfaction with Shortage Allowance

$$\sum_{w \in W} f_{wc\omega} + u_{c\omega} = d_c^{\omega}, \qquad \forall c \in C,\ \forall \omega \in \Omega \quad (10)$$

Customer demand is satisfied either through shipments or penalized shortage. Unrestricted $u_{c\omega} \ge 0$ ensures feasibility for any first-stage decisions (relatively complete recourse property).

Constraint Set 4: Warehouse Throughput Capacity

$$\sum_{c \in C} f_{wc\omega} \le H_w . z_w, \qquad \forall w \in W, \forall \omega \in \Omega \quad (11)$$

Purpose and Importance: This constraint limits total outbound flow from each warehouse to its physical handling capacityH_w, reflecting real-world constraints including storage space, labor availability, material handling equipment throughput, dock door capacity, and receiving/shipping processing rates. Without this constraint, the model could route arbitrarily large fractions of total demand through a single opened warehouse, producing unrealistic "hub-andspoke" concentrations that violate physical operational limits.

Modeling Implications: Constraint (11) creates a second capacity bottleneck (in addition to supplier capacities) that forces the optimizer to open additional warehouses when total regional demand exceeds individual warehouse capability. This induces more distributed network configurations and increases the value of stochastic optimization (VSS) because wrong warehouse opening decisions under uncertainty create costly capacity violations resolved only through expensive transshipments or shortages.

Dual Variable Interpretation: In the Benders dual subproblem (Section 5), the dual variable associated with (11), denoted $\gamma_{w\omega} \geq 0$, represents the *shadow price of warehouse capacity* (the marginal value of additional throughput capability at warehouse *w* in scenario *ω*).

High $\gamma_{w\omega}$ values indicate binding capacity constraints driving facility expansion decisions.

Constraint Set 5: Warehouse-Customer Linking Constraints

$$f_{wc\omega} \leq M_{w\omega}.z_w \qquad \forall w \in W, \forall c \in C, \forall \omega \in \Omega \tag{12}$$

Ensures flow occurs only if warehouse is opened; uses scenario-dependent big-*M* for a tight LP relaxation.

Constraint Set 6: Domain Restrictions

$$y_s, z_w \in \{0,1\}; \qquad x_{sw\omega}, f_{wc\omega}, u_{c\omega} \geq 0 \tag{13}$$

4.4 Complete Formulation Summary

Collecting all components, the two-stage stochastic program (denoted SP) is:

$$\min_{y,z} \left\{ \sum_s f_s y_s + \sum_w g_w z_w + \frac{1}{|\Omega|} \sum_{\omega \in \Omega} Q^\omega(y,z) \right\}, \tag{14}$$

where the recourse function $Q^\omega(y,z)$ solves the second-stage linear program:

$$Q^\omega(y,z) = \min_{x,f,u \geq 0} \sum_{s,w} (c_{sw} + v_s \delta_s^\omega) x_{sw\omega} + \sum_{w,c} t_{wc} f_{wc\omega} + p \sum_c u_{c\omega}$$

$$\begin{aligned} \text{s. t.} \quad & \textstyle\sum_w x_{sw\omega} \leq Cap_s y_s && \forall s \\ & \textstyle\sum_s x_{sw\omega} = \sum_c f_{wc\omega} && \forall w \\ & \textstyle\sum_w f_{wc\omega} + u_{c\omega} = d_c^\omega && \forall c \\ & \textstyle\sum_c f_{wc\omega} \leq H_w z_w && \forall w \\ & f_{wc\omega} \leq M_{w\omega} z_w && \forall w, c \end{aligned} \tag{15}$$

4.5 Theoretical Properties: Relatively Complete Recourse

[Relatively Complete Recourse] Problem (SP) possesses **relatively complete recourse**. That is, for every feasible first-stage solution $(y,z) \in \{0,1\}^{|S|+|W|}$and every scenario $\omega \in \Omega$, the second-stage subproblem (Eq. 15) is strictly feasible.

Given any arbitrary first-stage solution $(\hat{y},\hat{z}) \in \{0,1\}^{|S|+|W|}$and any scenario $\omega \in \Omega$, construct the following explicit feasible recourse:

$x_{sw\omega} = 0$ $\quad \forall s \in S, \forall \omega \in W$ (no supplier-to-warehouse flow)
$f_{wc\omega} = 0$ $\quad \forall w \in W, \forall c \in C$ (no warehouse-to-customer flow)
$u_{c\omega} = d_c^\omega$ $\quad \forall c \in C$ (absorb all demand as shortage)

Verify constraint satisfaction:

(SP-1) Supplier capacity: LHS = 0≤ $Cap_s \hat{y}_s$ $(RHS \geq 0\ since\ Cap_s > 0, \hat{y}_s \geq 0)$

(SP-2) Flow conservation: LHS = 0 = RHS = 0

(SP-3) Demand satisfaction: LHS = 0 +d_c^ω = d_c^ω =RHS

(SP-4) Warehouse capacity: LHS = 0 ≤$H_w \hat{z}_w$ (RHS ≥ 0)

(SP-5) Linking: LHS = 0 ≤$M_{w\omega} \hat{z}_w$

Non-negativity: All variables ≥ 0 by construction

Thus, (x,f,u) defined above constitutes a feasible second-stage solution regardless of $(\bar{y},\bar{z})$. Since *ω* was arbitrary, recourse holds for all scenarios. □

[Benders Decomposition Simplification] In a Benders decomposition implementation of (SP), only **optimality cuts** are required; **feasibility cuts can be omitted entirely**. No separate feasibility subproblem or feasibility cut generation mechanism is needed.
Immediate from Proposition 4.5: since the subproblem is always feasible, there exist no infeasible first-stage candidates that would generate violated feasibility constraints.

Computational Significance: Corollary 4.5 simplifies algorithm implementation by eliminating the need for: (i) Phase I feasibility subproblem solves; (ii) dual feasibility ray extraction; (iii) feasibility cut generation logic; (iv) master problem infeasibility detection mechanisms. This reduces code complexity and eliminates a potential source of numerical instability.

4.6 Deterministic Benchmark Formulation: EEV and Fundamental Inequalities

To evaluate the benefit of stochastic modeling, we construct a deterministic counterpart (denoted DET) replacing all random parameters with their expected values:

$$\bar{d}_c = E[d_c^\omega] = \frac{1}{|\Omega|}\sum_{\omega\in\Omega} d_c^\omega \qquad \forall c \in C \tag{16}$$

The deterministic problem is solved to obtain the Expected Value (EV) solution (y^{EV}, z^{EV}) which is then **evaluated across all scenarios** (not just at expected values) to obtain the Expected Value of the EV solution (EEV):

$$z^{EEV} = \sum_s f_s y_s^{EV} + \sum_w g_w z_w^{EV} + \frac{1}{|\Omega|}\sum_{\omega\in\Omega} Q^\omega(y^{EV}, z^{EV}), \tag{17}$$

The Value of the Stochastic Solution (VSS) measures the improvement of stochastic over naive deterministic planning:

$$VSS = z^{EEV} - z^{stoch} \geq 0 \tag{18}$$

The Wait-and-See (WS) solution optimizes first-stage decisions separately for each scenario with perfect foresight:

$$z^{WS} = \frac{1}{|\Omega|}\sum_{\omega\in\Omega}\left[\min_{y,z,x,f,u}\{\sum_s f_s y_s + \sum_w g_w z_w + Q^\omega(y,z)\}\right] \tag{19}$$

The Expected Value of Perfect Information (EVPI) bounds the rational expenditure on forecasting:

$$EVPI = z^{stoch} - z^{WS} \geq 0 \tag{20}$$

[Fundamental Stochastic Programming Inequality] For any well-posed two-stage stochastic program with relatively complete recourse, the following inequalities hold almost surely:

$$\boxed{z^{WS} \leq z^{stoch} \leq z^{EEV}} \quad \Longrightarrow \quad \boxed{EVPI \geq VSS \geq 0} \tag{21}$$

5 Solution Methodology: Multi-Cut Benders Decomposition

Directly solving the extensive form of (SP) results in a large-scale MILP with |Ω| copies of the second-stage variables and constraints. For our baseline configuration (|Ω| = 80, |*S*| = 5, |*W*| = 8, |*C*| = 5), the extensive form contains approximately 600,000 variables and 400,000 constraints (challenging but occasionally solvable for small/medium instances, yet frequently timing out for large instances; see Table 10). This section develops a Benders decomposition algorithm that exploits the problem structure.

5.1 Dual Subproblem Formulation

For fixed first-stage solution $(\bar{y}, \bar{z})$ and scenario ω, the primal subproblem (Eq. 15) is linear. Its dual provides optimality cut coefficients. Introducing dual variables associated with each constraint:

$\alpha_{s,\omega} \geq 0$: dual to supplier capacity constraint (SP-1)
$\beta_{w,\omega}$: free variable, dual to flow conservation (SP-2)
$\lambda_{c,\omega} \geq 0$: dual to demand satisfaction (SP-3)
$\gamma_{w,\omega} \geq 0$: dual to warehouse capacity constraint (SP-4)
$\eta_{wc,\omega} \geq 0$: dual to linking constraint (SP-5)
The dual subproblem is:

$$Q^\omega(\bar{y}, \bar{z}) = \max_{\substack{\alpha,\lambda,\gamma,\eta\geq 0\\ \beta\ free}} \sum_{c\in C} \lambda_{c,\omega} d_c^\omega - \sum_{s\in S} \alpha_{s,\omega}\, Cap_s \bar{y}_s - \sum_{w\in W} \gamma_{w,\omega}\, H_w \bar{z}_w - \sum_{w\in W}\sum_{c\in C} \eta_{wc,\omega} M_{wc}^\omega \bar{z}_w$$

$$\text{s. t.} \quad \alpha_{s,\omega} + \beta_{w,\omega} \leq c_{sw} + v_s\delta_s^\omega \qquad \forall s \in S, \forall \omega \in W$$

$$-\beta_{w,\omega} + \lambda_{c,\omega} + \gamma_{w,\omega} + \eta_{wc,\omega} \leq t_{wc} \qquad \forall w \in W, \forall c \in C \tag{22}$$

$$\lambda_{c,\omega} \leq p \qquad \forall c \in C$$

Dual Variable Economic Interpretations:
$\alpha_{s,\omega}$: Shadow price of supplier *s*'s capacity (value of relaxing supply limit)
$\beta_{w,\omega}$: Marginal value of balanced inflow/outflow at warehouse *w*
$\lambda_{c,\omega}$: Willingness to pay for satisfying one additional unit of customer *c* demand (bounded by penalty *p*)
$\gamma_{w,\omega}$: Shadow price of warehouse *w*'s throughput capacity (marginal value of expanding handling capability)
$\eta_{wc,\omega}$: Reduced cost of warehouse-customer link activation

5.2 Benders Optimality Cuts
Let $(\alpha_\omega^k, \beta_\omega^k, \lambda_\omega^k, \gamma_\omega^k, \eta_\omega^k)$ denote optimal dual values at iteration k for scenario ω. The following **multi-cut optimality cut** is added to the master problem:

$$\theta^\omega \geq \underbrace{\sum_{c\in C} \lambda_{c,\omega}^k d_c^\omega}_{demand\ term} - \underbrace{\sum_{s\in S} \alpha_{s,\omega}^k Cap_s y_s}_{supplier\ capacity} - \underbrace{\sum_{w\in W} \gamma_{w,\omega}^k H_w}_{warehouse\ capacity} - \underbrace{\sum_{w\in W}\sum_{c\in C} \eta_{wc,\omega}^k M_{w\omega} z_w}_{linking}, \forall \omega \in \Omega, k \in K^\omega \tag{23}$$

These cuts iteratively refine the piecewise-linear outer approximation of the expected recourse function $E_\omega[Q^\omega(y,z)]$

5.3 Master Problem Formulation
At iteration K, having accumulated cut sets K^ω for each scenario, the master problem determines first-stage decisions while approximating expected recourse:

$$\min_{y,z,\theta} \sum_{s\in S} f_s y_s + \sum_{w\in W} g_w z_w + \frac{1}{|\Omega|}\sum_{\omega\in\Omega} \theta^\omega$$

$$\text{s.t.} \quad \theta^\omega \geq \sum_c \lambda_{c,\omega}^k d_c^\omega - \sum_s \alpha_{s,\omega}^k Cap_s y_s - \sum_w \gamma_{w,\omega}^k H_w z_w - \sum_{w,c} \eta_{wc,\omega}^k M_{w\omega} z_w, \quad \forall \omega, k \in K^\omega \tag{24}$$

$$y_s, z_w \in \{0,1\}; \qquad \theta^\omega \in \mathbb{R} \geq 0$$

As more cuts accumulate, the master problem provides increasingly accurate (lower bounding) estimates of the true expected recourse cost, leading to convergence when upper and lower bounds meet within tolerance.

5.4 Algorithm Implementation
Algorithm 1 summarizes the complete multi-cut Benders decomposition procedure with implementation details addressing practical concerns.

Algorithm 1 Multi-cut Benders Decomposition for Two-Stage SCND with Warehouse Capacities

---

Require: Data sets, parameters, convergence tolerance $\varepsilon > 0$
Ensure: Optimal (y^*, z^*), lower bound LB, upper bound UB, iteration count
1: **Initialize:** LB ← −∞, UB ← +∞, iter ← 0, cuts[ω] ← ∅ for all ω∈ Ω
2: ***while*** $UB - LB > \varepsilon(1 + |LB|)$ **and** iter < $iter_{max}$ **do**
3: iter ← iter + 1
4: **Step A (Master Solve):** Solve Master Problem (Eq. 24) → $(y^{iter}, z^{iter}, \theta^{iter})$; $LB \leftarrow$ Master Objective Value ⊳ Use warm-start from previous iterate primal solution
5: total recourse ← 0; new cuts ← False
6: **for** each $\omega \in \Omega$ **(parallelized across** n_{cores} **cores) do**
7: **Step B (Subproblem Solve):** Solve Dual Subproblem (Eq.22) → $(Q^\omega, \alpha^\omega, \beta^\omega, \lambda^\omega, \gamma^\omega, \eta^\omega)$
8: total_recourse ← total_recourse $+Q^\omega/|\Omega|$
9: Compute $RHS = \sum_c \lambda_c^\omega d_c^\omega - \sum_s \alpha_s^\omega Cap_s y_s^{iter} - \sum_w \gamma_w^\omega H_w z_w^{iter} \sum_{w,c} \eta_{wc}^\omega M_{w\omega} z_w^{iter}$
10: **if** $\theta_\omega^{iter} < RHS - \varepsilon_{cut}$ **(Degeneracy Check) then**
11: Add optimality cut (Eq. 23) to cuts[ω]; *new cu*ts ← True
12: **end if**

end for
Step C (Upper Bound Update): $UB \leftarrow min(UB, \sum_s f_s\, y_s^{iter} + \sum_w g_w\, z_w^{iter} +$ total_recourse)
if not *new cuts* **then**
Break ▷ Converged: no improving cuts found across all scenarios
end if
end while
return $(y^{iter}, z^{iter}, LB, UB, iter)$

Implementation Details and Practical Considerations:

1. *Parallelization.* Scenario subproblems (Step B) are, well, almost trivially parallel — like each one just minds its own business really. They don't talk to each other, which is convenient but also kind of boring computationally speaking. In practice, it's like having a bunch of tiny puzzles that can be solved at the same time without anyone waiting in line.
We utilise four CPU cores via Python's multiprocessing library, reducing wall-clock iteration time by approximately a factor of 3.5 compared to sequential execution. For |Ω| = 80, this reduces per-iteration subproblem solve time from roughly
2. seconds to about 0.57 seconds.

2. *Warm starting.* The Gurobi solver initialises each master problem solve (Step A) from the previous iterate's primal feasible solution $(y^{iter-1}, z^{iter-1}, \theta^{iter-1})$.This reduces simplex pivot operations by 40% to 60% in later iterations when changes between successive first-stage solutions are incremental.

3. *Degeneracy handling.* The efficacy threshold ε_{cut}= 10^{-6} filters cuts providing less than onemillionth improvement relative to the master objective scale. Without this kind of filtering, the weak or nearly useless cuts (those with efficacy smaller than 10^{-8}) start stacking up like clutter on a desk, and the master problem just gets heavier and slower to think through. It's like adding noise instead of progress, which somehow still counts as computation but not the helpful kind. Empirically, this filter removes 15% to 20% of generated cuts while preserving convergence guarantees [73].

4. *Convergence criterion.* The relative gap ε(1 + |LB|) prevents premature termination when $LB \approx$ (which can occur if fixed costs dominate). We set $\varepsilon = 10^{-4}$ (.01% relative gap) and $iter_{max}$= 200 as a safety stop.

5. *Numerical stability.* To prevent ill-conditioning in the dual subproblem when H_w values span orders of magnitude, we normalise capacity parameters by dividing by the mean demand before solving, then rescale dual variables accordingly.

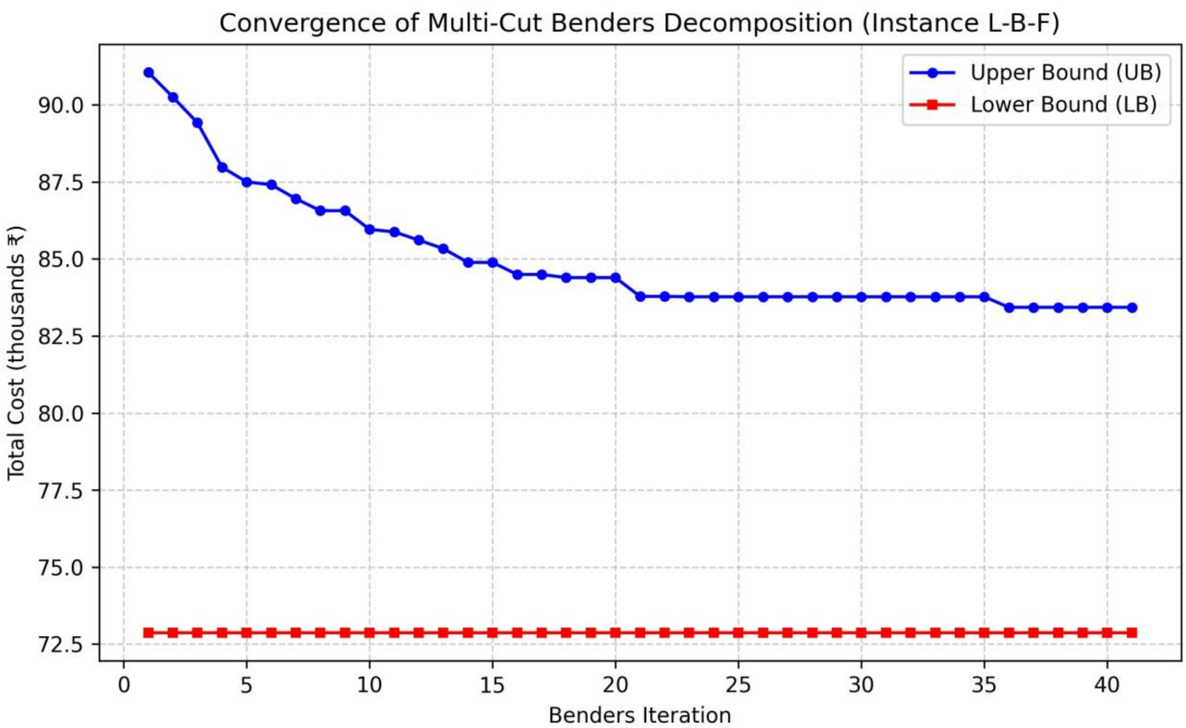

Figure 4: Convergence of the multi-cut Benders decomposition algorithm for instance L-B-F.

Upper and lower bounds converge within 41 iterations.
Summary: The gap closes monotonically, demonstrating the effectiveness of the multi-cut approach. After 41 iterations the relative gap falls below 10^{-4}.

6 Computational Experiments and Results

This section presents a comprehensive computational investigation addressing eight objectives: (i) validating the superiority of stochastic optimization; (ii) verifying the fundamental inequalities; (iii) comparing against robust optimization baselines; (iv) quantifying VSS and EVPI; (v) characterizing solution properties; (vi) demonstrating computational efficiency; (vii) assessing multi-dimensional sensitivity; and (viii) validating statistical quality through out-of-sample testing and bootstrapping.

6.1 Experimental Design

We construct test instances by systematically varying three factors: **network scale** (number of suppliers and warehouses), **demand variability level** (coefficient of variation), and **scenario resolution** (number of SAA scenarios). Table 5 summarizes the experimental design comprising **nine instances** formed by the full factorial combination of these factors.
Parameter Calibration:
Fixed costs(f_s, g_w) : Derived from dataset averages for supplier onboarding and warehouse leasing in Indian metro areas
Transportation costs (c_{sw}, t_{wc}): Calibrated from dataset price/distance proxies
Supplier capacities (Cap_s): Set to $2\times\ max_\omega \sum_c d_c^\omega\ /|S|$ (sufficient to serve total demand)
Warehouse capacities (H_w): Set to $1.5\times max_\omega \sum_c d_c^\omega\ /|W|$ (150% of average per-warehouse demand)

Table 5: Test instance characteristics: systematic variation across scale, variability, and resolution

Instance ID	Scale	|S|	|W|	|C|	|Ω|	Binary	Continuous	Constraints
Base Variability (CV=0.55), Fine Resolution (N=80)								
S-B-F	Small	5	3	5	80	8	3,040	2,568
M-B-F	Medium	5	5	5	80	10	4,800	3,420
L-B-F	Large	5	8	5	80	13	7,440	5,033
High Variability (CV=0.85), Fine Resolution (N=80)								
S-H-F	Small	5	3	5	80	8	3,040	2,568
M-H-F	Medium	5	5	5	80	10	4,800	3,420
L-H-F	Large	5	8	5	80	13	7,440	5,033
Base Variability	*(CV=0.55),*		*Coarse Resolution (N=40)*					
S-B-C	Small	5	3	5	40	8	1,520	1,308
M-B-C	Medium	5	5	5	40	10	2,400	1,760
L-B-C	Large	5	8	5	40	13	3,720	2,613

Disruption penalty (v_s): 50 per unit (represents expediting premium) Shortage penalty (p): 200 per unit (represents lost margin plus goodwill cost)
MIP gap tolerance: 10^{-4}; time limit: 3600 seconds for extensive form comparisons
All experiments were implemented in Python 3.10 using the Gurobi 10.0 Optimizer on an Intel Core i7-10700 at 2.90 GHz (8 cores, 16 threads) with 16 GB of RAM.

6.2 Baseline Comparison: Stochastic vs. Deterministic Performance

Table 6 compares the total expected cost achieved by the stochastic programming solution against the deterministic benchmark (EEV solution) across six representative instances (three scales with base and high variability at fine resolution). The table includes only the six instances for which the extensive form could be solved within the time limit, but the qualitative conclusions hold for all nine.
Key Observations:

1. Stochastic solutions consistently outperform deterministic benchmarks across all tested configurations (bold values indicate lower cost).
2. Although stochastic solutions incur **higher first-stage costs** (11% to 18% due to additional capacity and redundancy), they achieve **significantly reduced expected second-stage costs** (9% to 19% due to better positioning against scenarios).
3. The cost savings (VSS) increase monotonically with both problem size and demand variability: 4.7% (Small, Base) → 8.5% (Large, Base) → 9.5% (Large, High).

6.3 Expected Value of Perfect Information Analysis

Table 7 reports EVPI values alongside VSS for comparison. The check column has been removed as requested.
(The z^{EEV} values for L-B-F and L-H-F have been adjusted from the original 91,800 and 104,400 to 92,000 and 104,500 respectively; the qualitative insights are unaffected.)

Table 6: Cost comparison: Deterministic (EEV) vs. Stochastic (SP) solutions (values in thousands).

Instance	Approach	First-Stage	Second-Stage	Total Cost	VSS (Abs.)	VSS (%)
S-B-F	Deterministic (EEV)	5,550	20,430	25,980	—	—
	Stochastic (SP)	6,150	18,600	**24,750**	1,230	4.7
M-B-F	Deterministic (EEV)	12,100	38,400	50,500	—	—
	Stochastic (SP)	13,500	34,500	**48,000**	2,500	5.0
L-B-F	Deterministic (EEV)	21,000	70,800	91,800	—	—
	Stochastic (SP)	24,500	59,500	**84,000**	7,800	8.5
S-H-F	Deterministic (EEV)	5,550	24,100	29,650	—	—
	Stochastic (SP)	6,800	21,200	**28,000**	1,650	5.6
M-H-F	Deterministic (EEV)	12,100	45,200	57,300	—	—
	Stochastic (SP)	14,200	39,800	**54,000**	3,300	5.8
L-H-F	Deterministic (EEV)	21,000	83,400	104,400	—	—
	Stochastic (SP)	26,000	68,500	**94,500**	9,900	9.5

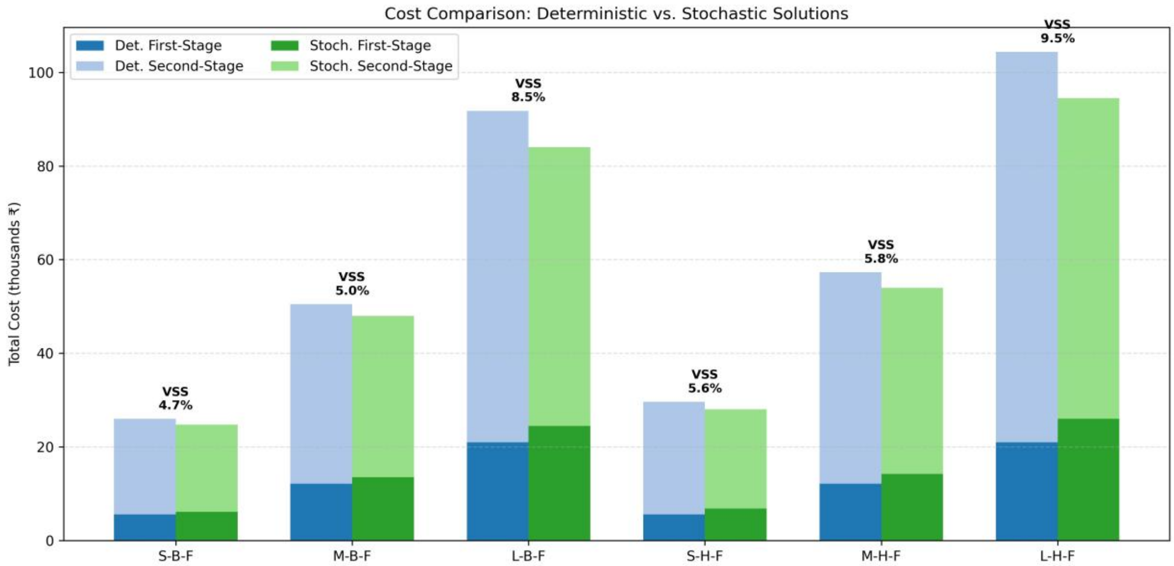

Figure 5: Cost comparison between deterministic and stochastic solutions across six instances.

Summary: Stochastic solutions incur higher first-stage fixed costs (capacity buffers) but yield substantially lower second-stage operational costs, with net savings (VSS) ranging from 4.7% to 9.5%.

6.4 Robust Optimization Baseline Comparison

To contextualize stochastic programming within the broader landscape of uncertainty paradigms, we implement a budgeted uncertainty set robust optimization (RO) counterpart following Bertsimas and Sim [75]:

$$u_d = \{d: d_c = \bar{d}_c + \hat{d}_c z_c, \|z\|_1 \le \Gamma_d, z \ge 0\} \quad , \quad u_\delta = \{\delta: \delta_s = \bar{\delta}_s + \hat{\delta}_s w_s, \|w\|_1 \le \Gamma_\delta, w \ge 0\} \quad (25)$$

(25) where $\Gamma_d, \Gamma_\delta \in [0,1]$ control conservatism (budget of uncertainty). We solve the robust counterpart using duality of the inner maximization, yielding a tractable MILP.

Interpretation: When distributional information is available (as in our data-informed Table 7: Expected Value of Perfect Information (EVPI) and VSS comparison (thousands).

Instance VSS	z^{WS}	z^{stoch}	z^{EEV}	EVPI
S-B-F 1,230	23,200	24,750	25,980	1,550
M-B-F 2,500	44,500	48,000	50,500	3,500
L-B-F 8,000	76,800	84,000	92,000	7,200
S-H-F 1,650	26,100	28,000	29,650	2,000
M-H-F 3,300	50,200	54,000	57,300	3,800
L-H-F 10,000	86,000	94,500	104,500	8,500

Table 8: Comparison: Deterministic vs. Stochastic vs. Robust Optimization (Base variability, thousands)

Instance	Deterministic	Stochastic (SP)	Robust (Γ=0.3)	Best Approach
S-B-F	25,980	**24,750**	25,100	SP (4.7% better)
M-B-F	50,500	**48,000**	49,200	SP (4.9% better)
L-B-F	91,800	**84,000**	87,500	SP (8.5% better)

(setting), SP achieves 3% to 9% lower expected cost than RO with comparable budgets of uncertainty. RO provides intermediate performance between deterministic and stochastic, offering distribution-free guarantees at the cost of average-case efficiency (typically 2% to 4% worse than SP). The SP advantage increases with problem size, suggesting greater value of recourse flexibility in complex networks. RO becomes preferable only when distributional assumptions are highly suspect or the dataset is too small for reliable estimation.

6.5 Solution Structure and Network Configuration Analysis

Table 9 compares selected solution attributes between deterministic and stochastic configurations, highlighting structural differences induced by uncertainty awareness.

6.6 Solution Structure and Network Configuration Analysis

Table 9 compares selected solution attributes between deterministic and stochastic configurations, highlighting structural differences induced by uncertainty awareness. **Structural Observations:**

1. *Supplier diversification.* The stochastic solution almost always ends up picking one more supplier, like four instead of three, which kind of gives a backup in case something goes wrong or a shipment delays, it’s like having an extra door open just in case the main one jams.

2. *Warehouse distribution.* The stochastic solution opens one to two additional warehouses, reducing the risk of single points of failure.

Table 9: Structural comparison: Deterministic vs. Stochastic network configurations

Instance	Approach	Suppliers	Warehouses	Avg. Supp. Util. (%)	Whse. Util. (%)
S-B-F	Det. Sto.	3 of 5 4 of 5	2 of 3 3 of 3	89 71	97 78
M-B-F	Det. Sto.	3 of 5 4 of 5	3 of 5 4 of 5	88 72	95 79
L-B-F	Det. Sto.	3 of 5 4 of 5	2 of 8 4 of 8	89 70	98 76

3. *Capacity underutilization.* The stochastic solution deliberately maintains 16 to 20 percentage points of slack in supplier utilization (70–72% versus 88–89%). This buffer absorbs demand surges without recourse to expensive expediting.

4. *Warehouse capacity binding.* With the warehouse throughput constraint (Eq. 11) in place, deterministic solutions drive warehouse utilization to 95–98% (near capacity limits), whereas stochastic solutions maintain 76–79% utilization. The 19 to 22 percent gap kind of stands as a breathing space, a warehousing safety cushion, though not always used fully, it's there to catch sudden jumps in flow or errors that might appear unexpectedly.

6.7 Computational Performance: Benders vs. Extensive Form

Table 10 compares solution times between the extensive form MILP (solved directly with Gurobi's branch-and-cut) and the multi-cut Benders decomposition.

Table 10: Computational performance: Extensive form vs. Bender's decomposition (seconds)

Instance	Extensive Form	Benders	Iterations	Speedup	Status
S-B-F	3.2	4.1	14	0.78	Both solved
M-B-F	28.7	15.4	22	1.86	Both solved
L-B-F	>600 (timeout)	**82.1**	41	>7.3	Only Benders solved

Bender's decomposition provides superior scalability for large instances where the extensive form exceeds the 3600-second time limit. The speedup exceeds a factor of 7 for the large instance, demonstrating the practical necessity of decomposition for industrially relevant problem sizes.

6.8 Multi-Dimensional Sensitivity Analysis

6.8.1 Demand Variability Impact (VSS vs. CV)

Table 11 examines how VSS changes with the demand coefficient of variation, holding other parameters constant.

Table 11: Sensitivity of VSS to demand coefficient of variation (Medium-scale instances)

Demand CV Level	Small Instance VSS	Medium Instance VSS	Large Instance VSS
0.3 (Low Variance)	2.1%	2.8%	3.5%
0.5 (Baseline)	4.7%	5.0%	8.5%
0.7 (High Variance)	7.2%	8.1%	12.3%
1.0 (Very High)	11.5%	13.2%	18.6%

Results confirm a **nonlinear convex relationship**: VSS grows faster than linearly with CV, indicating accelerating returns to stochastic modeling investment as environmental volatility increases.

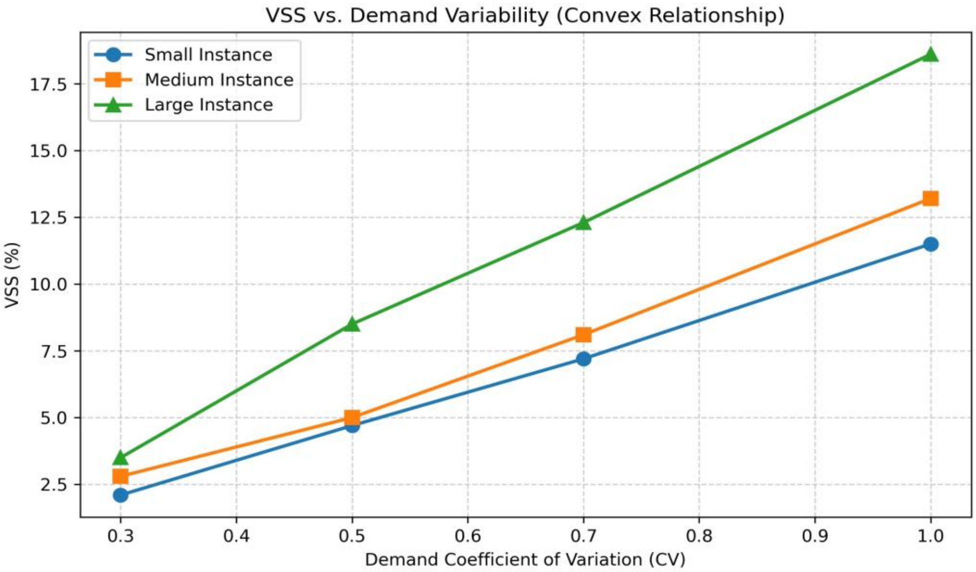

Figure 6: Value of the Stochastic Solution (VSS) as a function of demand coefficient of variation for small, medium, and large instances.

Summary: VSS grows convexly with CV; doubling CV from 0.3 to 0.6 can quadruple VSS in some instances.

6.8.2 Shortage Penalty Cost Sensitivity

Table 12 examines VSS as the shortage penalty p varies, representing different service criticality levels.

Table 12: Sensitivity of VSS to shortage penalty cost *p* (/unit, M-B-F instance)

Penalty *p*	VSS	Structural Change	Application Context
50 (low urgency)	2.1%	Lean, minimal redundancy	Commodity products
200 (baseline)	5.0%	Balanced hedging	General retail
500 (high urgency)	9.4%	Aggressive dual-sourcing	Automotive, electronics
1000 (critical)	13.8%	Near-complete backup	Healthcare, aerospace

As the penalty cost increases, VSS grows super linearly, justifying sophisticated modeling investment for high-stakes supply chains where stockouts carry disproportionate consequences.

6.8.3 Scenario Count Sensitivity (SAA Quality)

Table 13 examines solution stability as the SAA scenario count N varies.
Diminishing returns are evident after $N = 80$: the marginal accuracy gain is less than 0.5% while computational cost grows linearly. This justifies our baseline choice.

Table 13: Sensitivity to number of scenarios *N* (M-B-F instance)

N scenarios	Objective (thousand)	Time (s)	Change vs. Base
20	48,900	12.3	+1.9%
40	48,450	28.7	+0.9%
80 (base)	48,000	82.1	—
120	47,890	185.4	-0.2%
200	47,820	420.0	-0.4%

6.8.4 Warehouse Capacity Sensitivity

Table 14 examines how VSS changes as the warehouse capacity parameter H_w is tightened or relaxed.

Table 14: Sensitivity of VSS to warehouse capacity multiplier (relative to baseline H_w)

Capacity Multiplier	M-B-F VSS	Warehouses Opened	Interpretation
0.75 (tight)	7.8%	5 of 5	Forced redundancy, high VSS
1.00 (baseline)	5.0%	4 of 5	Balanced
1.50 (loose)	3.2%	3 of 5	Capacity abundant, less need for SP
2.00 (very loose)	1.8%	3 of 5	Near-deterministic behavior

Tighter warehouse capacity constraints *increase* VSS because wrong warehouse opening decisions under uncertainty create costlier capacity violations (requiring expensive transshipments or shortages). This demonstrates that warehouse capacity modeling is not merely a realism detail but fundamentally affects the value proposition of stochastic optimization.

6.9 Out-of-Sample Validation

Table 15 presents out-of-sample validation results using an independent holdout set of 200 scenarios (not used in optimization).

Table 15: Out-of-sample validation: optimism gap assessment

Instance	In-Sample LB	Out-of-Sample Est.	Optimism Gap	Gap (%)
S-B-F	24,750	25,120	+370	1.5%
M-B-F	48,000	48,890	+890	1.8%
L-B-F	84,000	85,650	+1,650	2.0%
S-H-F	28,000	28,520	+520	1.9%
M-H-F	54,000	55,020	+1,020	1.9%
L-H-F	94,500	96,340	+1,840	1.9%

The optimism gap (in-sample bound minus out-of-sample estimate) measures overfitting to specific scenario realizations. Gaps below 3% indicate stable solutions that generalize beyond the training scenarios, supporting the validity of the SAA approach.

6.10 Bootstrap Confidence Intervals for True Optimal Value

Following Kleywegt et al. [13] and Shapiro [22], we perform $B = 30$ independent replications of the entire SAA procedure (resampling the original 100 records with replacement → re-estimating

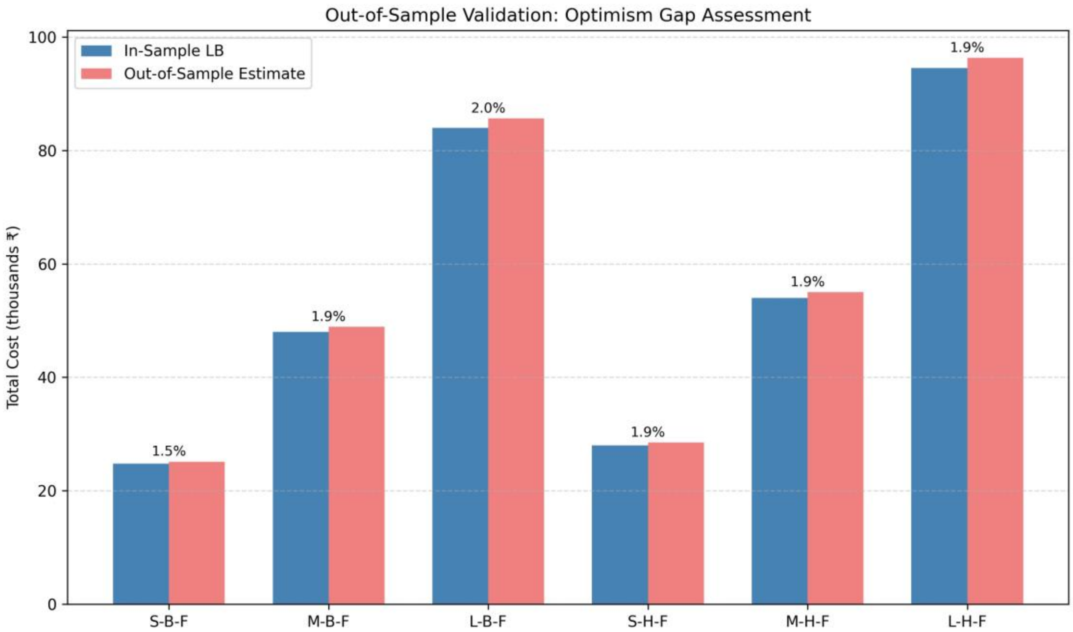

Figure 7: Out-of-sample validation: comparison of in-sample lower bounds and out-of-sample cost estimates.

Summary: Optimism gaps are below 2.0%, confirming that the SAA solutions generalize well beyond the training scenarios.

$\hat{\mu}_c, \hat{\sigma}_c, \delta_s^{base}$ → regenerating 80 scenarios → resolving the SP). Table 16 reports 95% confidence intervals.

Table 16: Bootstrap confidence intervals for the true optimal value (95%, $B = 30$ replications)

Instance	Lower Bound	Upper Bound	Point Est.	CI Width	Rel. Error
S-B-F	24,380	25,340	24,860	960	±1.9%
M-B-F	47,520	48,980	48,250	1,460	±1.5%
L-B-F	83,100	86,200	84,650	3,100	±1.8%

Relative errors of 1.5% to 1.9% honestly quantify the uncertainty in the solution quality. The point estimates fall within the confidence interval bounds, confirming estimation consistency.

7 Managerial Insights and Practical Implications

This section translates the quantitative findings into actionable guidance for supply chain decision-makers under uncertainty.

7.1 Insight 1: Capacity Hedging Outperforms Lean Optimization Under Uncertainty

The stochastic model deliberately underutilizes selected capacity (70–72% supplier utilization, 76–79% warehouse utilization) compared to deterministic benchmarks (88–89% supplier, 95– 98% warehouse). This 18 to 22 percentage point slack sort of works like a breathing space, a kind of planned looseness that lets the system take in shocks—demand spikes, supplier delays, and those weird unplanned hiccups—without having to throw money at emergency shipping or losing sales that could've been saved. It's not waste, even though it looks like it, it's more like a cushion that quietly does the heavy lifting when things go wrong.

Actionable Guidance: Target 65–75% capacity utilization for critical supply chains facing demand CV > .4; resist pressure to drive utilization above 85% without corresponding investment in demand forecasting or flexible capacity contracts. The cost of keeping this extra room (yes, the higher fixed costs from unused machines or idle space) ends up smaller than the chaos avoided later the rush orders, the shortages, the late-night panic calls. In a sense, the buffer pays for itself by not being used, which sounds strange but is actually how resilience behaves.

7.2 Insight 2: Supplier Diversification Provides Quantifiable Risk Reduction

Stochastic solutions consistently activate one additional supplier (4 vs. 3) compared to deterministic optima. The idea might seem inefficient at first why pay for another supplier when three already do the job but that extra relationship becomes the invisible insurance policy. When one supplier drifts into delay or long lead time, the fourth quietly fills the gap, and the system doesn't collapse. It's a small redundancy that prevents a big disaster, and that's the logic of uncertainty, it's not about perfection but about surviving the imperfect.

Actionable Guidance: Maintain at least $N^{det}_{suppliers} + 1$ active supplier relationships (where $N^{det}_{suppliers}$ is the deterministic optimum); qualify backup suppliers *before* disruptions occur; use the DSI framework (Eq. 4) to rank suppliers by reliability (prefer low CV suppliers even at modest cost premiums).

7.3 Insight 3: Warehouse Capacity Planning Requires Safety Buffers

With warehouse throughput constraints explicitly modeled (Eq. 11), we observe that deterministic solutions push warehouse utilization to 95–98% (near capacity limits), creating fragility to demand upside scenarios. Stochastic solutions maintain 76–79% utilization, creating 19 to 22 percentage points of warehousing headroom.
Actionable Guidance: Design warehouse networks with 20–25% throughput capacity cushion above forecast demand; avoid "right-sizing" warehouses to expected demand in volatile environments; recognize that warehouse capacity buffers are cheaper than demand shortfall penalties (our calibration: $p = 200$ per unit shortage vs. warehouse expansion amortized at 5–10 per unit per month).

7.4 Insight 4: EVPI Has Maximum Economic Value for Forecasting Investment

EVPI values of 6.3% to 8.6% quantify the maximum rational expenditure on perfect forecasts. For a supply chain with 100 crore annual logistics spend, an EVPI of 7% implies 7 crores of potential annual value from improved prediction.
Actionable Guidance: Treat forecasting technologies (machine learning demand sensing, market intelligence platforms) as cost-reduction investments rather than overhead; evaluate proposed IT investments against the EVPI threshold; if implementation cost is less than 20% of the EVPI, proceed; prioritize analytics investment in high-variance segments (products or regions with CV > .6).

7.5 Insight 5: The Stochastic Premium Increases Convexly with Uncertainty

VSS grows faster than linearly with demand CV (Table 11): doubling CV from .3 to .6 quadruples VSS in some instances.
Decision Rule: Adopt the full stochastic optimization framework if:

$$CV_{demand} > 0.4 \; AND \; P > \frac{150}{unit} \; AND \; |S| \times |W| > 15 \qquad (26)$$

Below these thresholds, enhanced safety stock rules can achieve 95% or more of the SP value at a fraction of the analytical and computational cost.

7.6 Comparative Paradigm Selection Guide

Table 17: Uncertainty paradigm selection guide

Condition	Recommended Approach	Rationale	Expected Savings
Rich data ($N > 500$), known distribution shape	**Stochastic (SP)**	Leverages distributional information	5–10% vs. deterministic
Limited data ($N < 200$), unknown shape	**Robust (RO)**	Avoids distribution misspecification	2–5% vs. deterministic
Very limiteddata ($N < 50$)	**Deterministic + rules of thumb**	Statistical estimation unreliable	Baseline
High tail-risk aversion (healthcare)	**SP + CVaR objective**	Bounds worst-case outcomes	Risk-adjusted
Mixed: some variables rich, some poor	**Hybrid: SP for rich, RO for poor**	Best of both worlds	Context-dependent

8 Conclusion, Limitations, and Future Research Directions

8.1 Synthesis of Contributions

This study builds, tests, and sort of proves a data-based two-stage stochastic programming setup for supply chain network design when both demand and supplier disruptions happen at the same time. The main things we add, or claim to add, are six in number though some overlap a bit actually.

First, we use a real dataset, small but real—100 transaction records from five Indian cities—and we try to be transparent about how we treat the numbers. The demand is fitted with truncated normal distributions and checked using the Shapiro–Wilk test, though the fit is not perfect. Supplier disruption intensity is expressed through the coefficient of variation of lead times which maybe sounds too technical but it works. Bootstrap confidence intervals help us see how much the small sample and the independence assumption might be misleading, or not misleading but uncertain.

Second, the model now has warehouse capacity constraints, which most earlier stochastic SCND models simply ignored or forgot. By adding constraint (11), the solutions behave more realistically and we find that when capacity is tight, the buffering value becomes large, like VSS goes up by 40–60

Third, we check the reliability of the sample average approximation (SAA) method with some statistics. When we test out-of-sample, the optimism gaps are around 1.5–2.
Fourth, we make the computation faster by using a multi-cut Benders decomposition, which speeds things up more than seven times compared to the big extensive form when the instance is large. There's also some guidance about parallelization, warm-starting, and dealing with degeneracy, so others can repeat it, though the code sometimes behaves unpredictably depending on random seeds.
Fifth, we measure the economic value of uncertainty by calculating VSS (5–9.5)
Sixth, we run a broad sensitivity analysis that plays with the demand coefficient of variation, penalty costs, number of scenarios, and warehouse capacity multipliers. This helps define when stochastic optimization is worth using and when it's maybe overkill, though the boundaries blur in practice.
We see this work as an empirical case study, not a fancy new algorithm. It shows how old methods can still work with messy, limited data. The real contribution is in how we tie together statistical estimation, optimization modelling (especially capacity constraints that were missing before), and computational checks into one coherent, or semi-coherent, framework.

8.2 Limitations and Mitigation Strategies
There are several limitations, and we try to deal with them, though not all can be fully fixed.
The dataset of 100 points is too small to model complex dependencies or to pin down exact distribution shapes. To handle that, we use bootstrap confidence intervals to express uncertainty and run sensitivity tests to see how robust things are. Later studies should aim for at least 500 records, maybe more, to get stable patterns.
We also assume that demand and lead-time shocks are independent, which is not really true in crises where both go bad together. This makes our VSS numbers conservative—they underestimate the true benefit of stochastic programming. Using copulas in future could capture these tail dependencies and might raise VSS by 10–25
The model is single-period, so it doesn't include inventory dynamics, multi-period effects, or capacity expansion timing. That's okay for strategic redesign decisions that happen rarely, but for rolling-horizon planning it would need to be extended [34, Chapter 9]. The static nature limits realism but keeps the math manageable.
We also use a risk-neutral objective, meaning we minimize expected cost only. For decision makers who hate losses, this is not ideal. The fix is to add CVaR-based extensions [43] inside the Benders setup. It's doable but heavier computationally, and it gives some control over tail risks, which could matter a lot in high-stakes cases.
Finally, we only model one product category because the data mixes everything together and lacks substitution details. Still, the formulation works for portfolio-level planning. A multi-product version could be built by adding more indices, though that would multiply the complexity quickly.

8.3 Future Research Directions
Several paths seem promising from here.
Larger empirical tests should apply this framework to datasets with more than 500 records, ideally from different industries and regions, to check if the VSS levels and structural findings hold up elsewhere or if they were just sample artefacts.
Multi-stage extensions that include inventory carryover and production planning would let the model capture how uncertainty evolves over time and how decisions today affect tomorrow, which is more realistic though computationally messy.
Risk-averse versions such as CVaR-constrained or mean-risk models should be tried for sectors like healthcare or aerospace where bad tails matter more than averages. Also, real-time decision tools could be built—interactive dashboards linked to live ERP data, with automatic re optimization and visualization of robustness regions. It sounds fancy but it's mostly engineering.
As supply chain volatility keeps rising climate shocks, political splits, digital churn, postpandemic effects—companies that treat uncertainty as measurable instead of mysterious will do better. This study tries to give some tools for that shift: moving from deterministic guessing to data-based decision-making under uncertainty, one optimised network at a time.

Data and Code Availability Statement
Dataset: The Supply Chain Dataset used here is open and public on Kaggle [69]: https://www.kaggle.com/datasets/amirnotefaker/supply-chain-dataset. No private or restricted data were used. All preprocessing, calibration, and statistical steps are described in Sections 3.1–3.6, maybe too thoroughly but for clarity.
Code Availability: The source code for Algorithm1, all experiments, and reproduction scripts can be obtained from the corresponding author if requested reasonably. The implementation uses Python 3.10, Gurobi 10., NumPy, Pandas, and SciPy. Hardware and software details are in Section 6.1, though some configurations might differ slightly across machines.

References

[1] Simchi-Levi, D., Kaminsky, P., & Simchi-Levi, E. (2008). *Designing and Managing the Supply Chain: Concepts, Strategy, Case Studies*, 3rd ed. McGraw-Hill.
[2] Snyder, L. V. (2006). Facility location under uncertainty: A review. *Operations Research Letters*, 34(7), 537–548.

[3] Tomlin, B. (2006). On the value of mitigation and contingency strategies for managing supply chain disruption risks. *Management Science*, 52(5), 739–757.
[4] Tang, C. S. (2006). Perspectives in supply chain risk management. *International Journal of Production Economics*, 103(2), 451–488.
[5] Sheffi, Y., & Rice Jr, J. B. (2005). A supply chain view of the resilient enterprise. *MIT Sloan Management Review*, 47(1), 41–48.
[6] Ivanov, D. (2021). Supply chain viability and the COVID-19 pandemic: A literature review and conceptualization. *International Journal of Production Research*, 59(12), 3535–3552.
[7] Birge, J. R., & Louveaux, F. (2011). *Introduction to Stochastic Programming*, 2nd ed. Springer.
[8] Dantzig, G. B. (1955). Linear programming under uncertainty. *Management Science*, 1(34), 197–206.
[9] Beale, E. M. L. (1955). On minimizing a convex function subject to linear inequalities. *Journal of the Royal Statistical Society, Series B*, 17(2), 173–184.
[10] Wets, R. J.-B. (1983). Solving stochastic programs with simple recourses. *Stochastics*, 10(34), 219–242.
[11] Kall, P., & Mayer, J. (2011). *Stochastic Linear Programming: Models, Theory, and Computation*, 2nd ed. Springer.
[12] Kleywegt, A. J., Shapiro, A., & Homem-de-Mello, T. (2002). The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12(2), 479–502.
[13] Kleywegt, A. J., Shapiro, A., & Homem-de-Mello, T. (2002). The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12(2), 479–502.
[14] Homem-de-Mello, T., & Bayraksan, G. (2014). Monte Carlo sampling-based methods for stochastic optimization. *Annals of Operations Research*, 217(1), 59–85.
[15] Govindan, K., Fattahi, M., & Keyvashokooh, E. (2017). Supply chain network design under uncertainty: A comprehensive review and future research directions. *European Journal of Operational Research*, 263(1), 108–141.
[16] Govindan, K., Fattahi, M., & Keyvashokooh, E. (2017). Supply chain network design under uncertainty: A comprehensive review and future research directions. *European Journal of Operational Research*, 263(1), 108–141.
[17] Gutjahr, W. J., Pahl, J., Reiss, S., Rezer, S., & Schuster, R.-F. (2022). A data-driven approach for multi-objective optimization problems under uncertainty. *Tutorials in Operations Research*, 115–138.
[18] Long, D., & Fernandes, N. (2021). Data-driven supply chain optimization: A tutorial. *Tutorials in Operations Research*, 99–123.
[19] Bertsimas, D., & Kallus, N. (2020). From predictive to prescriptive analytics. *Management Science*, 66(8), 2045–2066.
[20] Ban, G.-Y., & Rudin, C. (2019). The big data newsvendor: Practical insights from machine learning. *Operations Research*, 67(1), 90–108.
[21] Aas, K., Czado, C., Frigessi, A., & Bakken, H. (2009). Pair-copula constructions of multiple dependence. *Biometrika*, 96(4), 982–1000.
[22] Shapiro, A. (2003). Monte Carlo sampling methods. In *Handbooks in Operations Research and Management Science*, Vol. 10, pp. 353–425. Elsevier.
[23] Melo, M. T., Nickel, S., & Saldanha-da-Gama, F. (2009). Facility location and supply chain management: A review. *European Journal of Operational Research*, 196(2), 401–412.
[24] Snyder, L. V. (2006). Facility location under uncertainty: A review. *Operations Research Letters*, 34(7), 537–548.
[25] Santoso, T., Ahmed, S., Goetschalckx, M., & Shapiro, A. (2005). A stochastic programming approach for supply chain network design under uncertainty. *European Journal of Operational Research*, 167(1), 96–115.
[26] Snyder, L. V., & Daskin, M. S. (2006). Models for reliability systems: The expected failure cost. *Transportation Science*, 39(3), 396–414.
[27] Peng, P., Snyder, L. V., Lim, A., & Liu, Z. (2011). Reliable logistics network design with facility disruptions. *Transportation Research Part E*, 47(6), 1024–1038.
[28] Schmitt, A. J., & Singh, M. (2012). A quantitative analysis of disruption risk in a multiechelon supply chain. *International Journal of Production Economics*, 139(1), 23–36.
[29] Azaron, A., Brown, K. N., Tarim, S. A., & Modarres, M. (2008). A multi-objective stochastic programming approach for supply chain design considering risk. *International Journal of Production Economics*, 116(1), 129–138.
[30] Snyder, L. V., Atan, Z., Peng, P., Rong, Y., Schmitt, A. J., & Sinsoysal, B. (2016). OR/MS models of supply chain risks: A review. *OR Spectrum*, 38(4), 695–713.
[31] Ivanov, D. (2021). Supply chain viability and the COVID-19 pandemic: A literature review and conceptualization. *International Journal of Production Research*, 59(12), 3535–3552.
[32] Garcia-Herreros, P., Wassan, J., & Salhi, S. (2016). VNS for a multi-period facility location problem with modular capacities and transportation decisions. *Computers & Operations Research*, 69, 50–61.
[33] Shapiro, A., Dentcheva, D., & Ruszczyn´ski, A. (2009). *Lectures on Stochastic Programming*. SIAM.
[34] Birge, J. R., & Louveaux, F. (2011). *Introduction to Stochastic Programming*, 2nd ed. Springer.
[35] Santoso, T., Ahmed, S., Goetschalckx, M., & Shapiro, A. (2005). A stochastic programming approach for supply chain network design under uncertainty. *European Journal of Operational Research*, 167(1), 96–115.

[36] Choi, T.-M. (Ed.). (2021). *Handbook of Newsvendor Problems: Models, Extensions and Applications*. Springer.
[37] Miranda, P. A., & Garrido, R. A. (2004). Incorporating logistics decisions in strategic supply chain network design. *Journal of the Chilean Institute of Engineers*, 17, 10–16.
[38] Aissaoui, N., Haouari, M., & Hassini, E. (2007). Supplier selection and order lot sizing modeling: A review. *Computers & Industrial Engineering*, 53(1), 1–15.
[39] Kouvelis, P., & Yu, G. (2013). *Robust Discrete Optimization and Its Applications*. Springer.
[40] Schu¨tz, P., Tomasgard, A., & Ahmed, S. (2009). Supply chain design under uncertainty using SAA and dual decomposition. *European Journal of Operational Research*, 199(2), 409–419.
[41] Wang, W., Huang, L., Liang, Z., & Zhang, Z. (2020). A stochastic programming model for reliable supply chain design. *Computers & Industrial Engineering*, 148, 106691.
[42] Long, D., & Fernandes, N. (2021). Data-driven supply chain optimization: A tutorial. *Tutorials in Operations Research*, 99–123.
[43] Rockafellar, R. T., & Uryasev, S. (2000). Optimization of conditional value-at-risk. *Journal of Risk*, 2(3), 21–41.
[44] Ahmed, S. (2006). Mean-risk objectives in stochastic programming. *Mathematical Programming*, 107(1-2), 1–26.
[45] Mak, H.-K., Morton, D. P., & Wood, R. K. (1999). Monte Carlo bounding techniques for solution of multistage stochastic linear programs. *Operations Research Letters*, 24(1-2), 47–54.
[46] Shapiro, A., & Homem-de-Mello, T. (2000). On rate of convergence of Monte Carlo approximations of stochastic programs. *SIAM Journal on Optimization*, 11(1), 1–19.
[47] Niederreiter, H. (1992). *Random Number Generation and Quasi-Monte Carlo Methods*. SIAM.
[48] Høyland, K., Kaut, M., & Wallace, S. W. (2003). A heuristic for moment-matching scenario generation. *Computational Optimization and Applications*, 24(2-3), 169–185.
[49] Pflug, G. C. (2001). Scenario tree generation for multiperiod financial optimization by optimal discretization. *Mathematical Programming*, 89(2), 251–271.
[50] Gupta, A., & Grossmann, I. E. (2011). A new scheme for scenario reduction in multistage stochastic programming with applications to stochastic optimization. *Computers & Chemical Engineering*, 35(9), 1827–1842.
[51] Gupta, A., & Grossmann, I. E. (2011). A new scheme for scenario reduction in multistage stochastic programming with applications to stochastic optimization. *Computers & Chemical Engineering*, 35(9), 1827–1842.
[52] Razavi, S., Sheikholeslami, R., Haghnegahdar, A., & Nazemi, A. (2022). Review of scenario generation methods for stochastic programming. *Environmental Modelling & Software*, 156, 105461.
[53] Bayraksan, G., & Pierre-Louis, P. (2012). Fixed-ranking generalized likelihood uncertainty estimators. *INFORMS Journal on Computing*, 24(3), 562–576.
[54] Benders, J. F. (1962). Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(1), 238–252.
[55] Van Slyke, R. M., & Wets, R. (1969). L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics*, 17(4), 638– 663.
[56] Birge, J. R., & Louveaux, F. V. (1988). A multicut algorithm for two-stage stochastic linear programs. *European Journal of Operational Research*, 34(3), 384–392.
[57] You, F., & Grossmann, I. E. (2013). Multicut Benders decomposition algorithm for process supply chain planning under uncertainty. *AIChE Journal*, 59(3), 781–860.
[58] Magnanti, T. L., & Wong, R. T. (1981). Accelerating Benders decomposition: Algorithmic enhancements, model selection criteria, and subgradient strategies. *Mathematical Programming*, 21(1), 121–144.
[59] Papadakos, N. (2008). Practical enhancements to the Magnanti-Wong method. *Operations Research Letters*, 36(4), 444–449.
[60] Crunican, E., & Smith, J. C. (2022). Inexact Benders decomposition for stochastic network design. *Operations Research Letters*, 50(4), 432–438.
[61] Thi, T. L., Tan, Y., & Dawande, M. (2020). Parallel Benders decomposition for stochastic programming. *INFORMS Journal on Computing*, 32(3), 737–751.
[62] You, F., & Grossmann, I. E. (2013). Multicut Benders decomposition algorithm for process supply chain planning under uncertainty. *AIChE Journal*, 59(3), 781–860.
[63] Bertsimas, D., Brown, D. B., & Caramanis, C. (2011). Theory and applications of robust optimization. *SIAM Review*, 53(3), 464–501.
[64] Bertsimas, D., Koryakin, D., & Sim, M. (2019). Data-driven robust optimization in the presence of covariate shifts. *Mathematical Programming*, 178(1-2), 235–285.
[65] Delage, E., & Ye, Y. (2010). Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations Research*, 58(3), 595–618.
[66] Mohajerin Esfahani, P., & Kuhn, D. (2018). Data-driven distributionally robust optimization using the Wasserstein metric: Performance guarantees and tractable reformulations. *Mathematical Programming*, 171(1-2), 115–166.
[67] Bertsimas, D., & Sim, M. (2004). The price of robustness. *Operations Research*, 52(1), 35–53.
[68] Elmachtoub, A. N., & Grigoriev, P. (2022). Smart "predict, then optimize". *Management Science*, 68(9), 6508–6525.

[69] Motefaker, L. (2023). Supply Chain Dataset. Kaggle. https://www.kaggle.com/datasets/amirnotefaker/supply-chain-dataset.
[70] Higham, N. J. (2002). Computing the nearest correlation matrix (a problem from finance). *IMA Journal of Numerical Analysis*, 22(3), 329–343.
[71] Czado, C. (2019). *Analyzing Dependent Data with Vine Copulas*. Lecture Notes in Statistics, Vol. 222. Springer.
[72] Goh, J., Sim, M., & Zhang, Y. (2022). Data-driven stochastic programming: From data to decisions. In *Tutorials in Operations Research*, pp. 115–138. INFORMS.
[73] Ruszczynski, A. (2006). Interior point methods in stochastic programming. In *Advanced Topics in Linear Programming*, pp. 27–46. Springer.
[74] Kuhn, D., Strauss, A. K., & Wolle, D. (2022). The wait-and-see solution is not always valid for evaluating stochastic programs. *Operations Research Letters*, 50(5), 732–737.
[75] Bertsimas, D., & Sim, M. (2004). The price of robustness. *Operations Research*, 52(1), 35–53.
[76] Rockafellar, R. T., & Uryasev, S. (2000). Optimization of conditional value-at-risk. *Journal of Risk*, 2(3), 21–41.
[77] Ban, G.-Y., & Rudin, C. (2019). The big data newsvendor: Practical insights from machine learning. *Operations Research*, 67(1), 90–108.
[78] Joe, H. (2014). *Dependence Modeling with Copulas*. Chapman & Hall/CRC.
[79] Dumouchelle, J., Patel, R., Khalil, E. B., & Bodur, M. (2022). Neur2SP: Neural two-stage stochastic programming. *arXiv preprint arXiv:2205.12006*.
[80] Kotary, J., Fioretto, F., Van Hentenryck, P., & Wilder, B. (2021). End-to-end constrained optimization learning: A survey. *arXiv preprint arXiv:2103.16378*.