



Resilient supply chain network design under super-disruption considering inter-arrival time dependency: a new data-driven stochastic optimization approach

Mohammad Mahdi Vali-Siar^{a,*}, Hamid Tikani^b, Emrah Demir^c, Yousof Shamstabar^d

^a School of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran

^b Department of Industrial Engineering, K. N. Toosi University of Technology, Tehran, Iran

^c Business & Economics Artificial Intelligence Research Network, Cardiff Business School, Cardiff University, Cardiff, UK

^d Department of Industrial Engineering, Faculty of Engineering, Kharazmi University, Tehran, Iran



ARTICLE INFO

Keywords:

Resilience
Super-disruptions
Supply chain network design
Machine learning
Hyper-matheuristic

ABSTRACT

During large-scale disruptions, particularly super-disruptions such as global pandemics or large-scale natural disasters, supply chains are exposed to significant adverse impacts. This paper addresses the resilience in a supply chain network design problem under disruption risk by explicitly modeling the dependency between the inter-arrival times of disruptive events and severity of their consequences. A novel data-driven stochastic optimization framework is proposed to consider the ripple effects that typically propagate across supply chain networks following severe disruptions. Specifically, we have devised a hybrid methodology that integrates a clustering algorithm (unsupervised machine learning technique), a phase-type disruption model, and a two-stage stochastic model. To elaborate, a genetic-based clustering algorithm is used to identify the structure dependencies in the input data. Phase-type distributions and their associated theorems are then used to determine the probability distributions of disruptions. A novel mathematical model is developed to design the supply chain using the scenarios generated based on the obtained distributions, which is then solved using the Lagrangian decomposition combined with a new hyper-matheuristic algorithm. The computational efficiency and practical value of the proposed approach are demonstrated through a real-world case study. The findings highlight the effectiveness of developed methodology in designing a resilient supply chain, the proposed resilience strategies substantially improve the supply chain's performance compared to a non-resilient approach.

1. Introduction

Supply chains (SCs) are threatened by a variety of risks that can be classified into operational and disruption risks in terms of impact severity (Ivanov, 2021). Operational risks arise from inherent uncertainties in the supply chain, such as fluctuations in supply, demand, and costs. These risks have a high probability of occurrence but generally cause low impact. In contrast, disruption risks are less

* Corresponding author.

E-mail addresses: valisiar@iust.ac.ir (M.M. Vali-Siar), hamid.tikani@email.kntu.ac.ir (H. Tikani), demire@cardiff.ac.uk (E. Demir), y.shamstabar@gmail.com (Y. Shamstabar).

frequent but can cause severe and widespread consequences (Tang, 2006). Disruptions may originate from a wide range of sources, including natural disasters such as earthquakes, hurricanes, floods, and pandemics, as well as human-made threats like terrorism, wars, and strikes, or technical failures such as equipment breakdowns and system malfunctions (Sabouhi et al., 2018). According to Ranjkesh et al. (2019), the duration between shocks, known as the inter-arrival time, influences the extent of damage inflicted upon a system. A shorter inter-arrival time between disruptive events gives the system, such as a supply chain, less opportunity to recover after each shock, which reduces the system's ability to return to its steady state and self-recover and increases the risk of greater damage from subsequent disruptions (Ranjkesh et al., 2019; Melnyk et al., 2009). This issue has been considered in the current research.

In recent years, SCs have become more complex and susceptible to disruptions (Namdar et al., 2018). Hurricane Katrina in the United States (2006), tsunami and earthquake in Japan (2011), Ebola outbreak in West Africa (2014), transportation disruption in the Suez Canal (2021), and Francis Scott Key bridge collapse in the United States (2024) are some examples of disruptions that severely impacted SCs. These events highlight the vulnerability of modern supply chains and the necessity of building resilience. A resilient supply chain can withstand disruption risks and restore its operations to the original or even improved state in the post-disruption period (Christopher and Peck, 2004).

Super-disruptions are characterized by their prolonged duration, unpredictable scale, and the simultaneous disruption of supply, demand, and logistics. These disruptions not only affect multiple facets of the supply chain but also create a ripple effect, where disruptions in one segment propagate throughout the network. As extensively discussed in the literature, the ripple effect leads to various negative consequences, including higher costs, lower revenues, delivery delays, raw material shortages, and loss of customers (Ivanov, 2021). This cascade of disruptions increases the frequency and severity of events, reducing the inter-arrival time between shocks and further complicating the recovery process. From early 2020, the Covid-19 pandemic, as the most recent global super-disruption, severely disrupted supply chains, production, logistics, and demand over the course of several months. Almost all SCs were negatively affected (Sawik, 2022).

To mitigate the adverse effects of disruptions and minimize their spread, resilience strategies must be applied. These strategies are generally categorized into two types: proactive and reactive. Proactive strategies are implemented in advance, regardless of the likelihood of a disruption, aiming to prepare the SC for potential uncertainties. In contrast, reactive strategies are deployed after a disruption occurs, with the goal of restoring the SC network to its pre-disruption state or transitioning it to a predetermined desirable state. Some resilience strategies may serve both proactive and reactive purposes, depending on when and why they are used (Hohenstein et al., 2015; Fattah et al., 2017). Among the most effective and widely adopted resilience strategies are multiple sourcing, using backup facilities, facility fortification, capacity expansion, and dual-channel distribution (Vali-Siar and Roghanian, 2022).

This research focuses on the design of a resilient supply chain network. This is the first study to explicitly consider the impact of inter-arrival times of disruptions and the ripple effect within the context of supply chain network design. A clustering algorithm is used as an unsupervised machine learning (ML) method to detect the dependency structure, and phase-type distributions and associated theorems are used to find the probability distributions of cumulative damages. Using this technique enables the consideration of the interdependency between inter-arrival times of disruptions, their damages, and the ripple effect. Subsequently, disruption scenarios are generated for use in the SCND model. To formulate the SCND problem, a two-stage stochastic optimization model is developed to maximize the total profit of the SC. Lagrangian decomposition (LD) and a new hyper-matheuristic (HMH) algorithm, which applies reinforcement learning (RL), are proposed for solving the studied problem.

The remainder of the paper is structured as follows. In Section 2, the literature review is presented. Section 3 introduces the resilient SCND problem and presents the associated mathematical formulation. Section 4 provides an in-depth discussion of the disruption modeling approach, with particular emphasis on the application of pH distributions. In Section 5, the proposed solution methodologies are detailed. Section 6 presents the case study, along with the computational experiments and a comprehensive analysis of the results. Finally, Section 7 concludes the paper and outlines potential directions for future research.

2. Literature review

This section consists of two parts. The first reviews resilient SCND and the second discusses the applications of ML techniques in resilient SCND.

2.1. Resilient supply chain network design

Because of the critical role that network design plays in establishing resilient SCs, researchers have widely investigated SCND in this context. Azad et al. (2013) addressed disruptions in transportation modes and distribution centers (DCs) by applying hardening strategies and goods sharing. Later, Salehi Sadghiani et al. (2015) studied retail SCND with random disruptions in suppliers and vehicles and used backup supply facilities to mitigate the risks of disruption. Hasani and Khosrojerdi (2016) focused on designing a resilient global SC facing disruptions in suppliers, manufacturers, and wholesalers. They applied several resilience strategies, including multiple sourcing, holding extra inventory, and production of semi-manufactured products. Jabbarzadeh et al. (2016) applied facility fortification strategy to mitigate disruptions, including interruptions in both supply and demand. Fattah et al. (2017) studied the SCND problem considering responsiveness and resilience under warehouse disruptions. They proposed facility fortification and revising the assignment of customers to SC facilities for increasing SC resilience. The authors have particularly paid attention to the recovery process of affected facilities after the disruption. Responsive and resilient SCND was also investigated in the work of Sabouhi et al. (2020). In another study, Sabouhi et al. (2018) investigated disruption risks for suppliers and operational risks. Fortification of suppliers, multiple sourcing, and keeping additional inventory were employed. Vali-Siar and Roghanian (2020) addressed resilient SC

network redesign problem. [Vali-Siar et al. \(2022\)](#) addressed resilient mixed open- and closed-loop SCND problem under competition. They suggested multiple sourcing, adding extra production capacity, dual-channel distribution, and pricing to augment SC resilience. [Alikhani et al. \(2023\)](#) developed an approach for choosing suitable resilience strategies in SCND under disruption. Fortification, direct shipping, reassignment, and some other strategies were considered in their research. Recently, [Yilmaz et al. \(2025\)](#) applied a two-stage stochastic programming model with a conditional value-at-risk (CVaR) measure and chance constraints to manage risk and uncertainty in medical supply chain networks. [Zheng et al. \(2025\)](#) addressed resilience in the manufacturing sector by considering supplier irreplaceability. Their approach integrates backup suppliers, multiple sourcing strategies, and network redesign to enhance the overall resilience of the supply chain.

Some researchers have jointly addressed sustainability and resilience in SCND problem. [Hasani et al. \(2021\)](#) investigated resilient SCND under interval uncertainty with a focus on environmental issues. Their approach addressed disruptions in suppliers and transportation modes by implementing four resilience strategies. Similarly, [Sazvar et al. \(2021\)](#) examined sustainable and resilient SCND, utilizing capacity planning to bolster supply chain resilience. [Vali-Siar and Roghanian \(2022\)](#) advanced this line of research by developing a mixed-integer linear programming (MILP) model for a sustainable, resilient, and responsive mixed open- and closed-loop network. Their model incorporated multiple sourcing, dual-channel distribution, facility fortification, and increased production capacity to address disruptions in both forward and reverse logistics. [Foroozesh et al. \(2023\)](#) considered circular economy concept for sustainable-resilient SC design for perishable products while accounting for the network's financial resources. [Tsao et al. \(2024\)](#) introduced a MILP model for sustainable and resilient SCND under trade credit conditions, with an objective function aimed at minimizing unsatisfied demand as a resilience measure. Most recently, [Nili et al \(2025\)](#) studied sustainable and resilient closed-loop SCND for floating solar photovoltaic systems, adopting backup supplier contracts as a strategy to mitigate disruption risks.

Recently, the ripple effect has attracted growing attention in SC management literature. [Sawik \(2022\)](#) developed a stochastic mixed-integer programming (MIP) model for optimizing SC operations in the presence of ripple effect. [Sindhwan et al. \(2023\)](#) considered ripple effect in pharmaceutical SC using a multi-level analysis, including mathematical optimization, simulation and Bayesian network. [Liu et al. \(2023\)](#) focused on selecting best supplier actions in a SC under ripple effect. They proposed an integrated approach combining Markov decision process, Bayesian network and a nonconvex MIP model. [Sawik \(2023\)](#) investigated the SC viability in the conditions of occurring ripple effect through a stochastic quadratic model. Notably, none of the mentioned papers have worked on resilient SC network design under ripple effect. Existing studies have overlooked the temporal dependency between disruption events, particularly the influence of inter-arrival time on the severity and propagation of ripple effects. Moreover, limited attention has been given to data-driven stochastic modeling approaches that can dynamically capture disruption patterns and integrate them into resilient SCND decisions. These gaps highlight the need for more comprehensive methodologies that incorporate both stochastic disruption dynamics and advanced optimization techniques in SCND.

2.2. Applications of ML methods in resilient SCND

ML methods have gained increasing attention in SC management research. [Belhadi et al. \(2022\)](#) for instance, introduced a multi-criteria decision-making framework powered by artificial intelligence (AI) algorithms to support SC resilience strategies, demonstrating the promise of ML in this domain. [Zamani et al. \(2023\)](#) conducted a systematic review of AI and big data analytics for supply chain resilience, while [Yang et al. \(2023\)](#) provided a focused review on the use of ML in supply chain risk management. [Esmaeili et al. \(2023\)](#) developed predictive models to enhance the resilience of biomass SCs by reducing the mismatches between biomass supply and demand arising from disruptive conditions. To this end, a routing algorithm was designed to integrate ML-based predictive analytics into the supply chain. [Hao and Demir \(2023\)](#) and [Hao and Demir \(2024\)](#) conducted systematic literature reviews complemented by expert interviews to explore the applicability of AI in SC decision-making contexts. [Camur et al. \(2024\)](#) presented ML-based prediction models to predict availability dates of products during disruptions, thereby aiding SC resilience. Additional studies by [Roozkhosh et al. \(2023\)](#) and [Alhasawi et al. \(2023\)](#) further explore various ML applications for enhancing SC resilience.

To the best of our knowledge, there is limited research on resilient SCND incorporating the applications of ML techniques. [Jabarzadeh et al. \(2018\)](#) presented a hybrid approach for a resilient and sustainable SCND problem under random disruptions affecting both suppliers and manufacturers. In their study, supplier sustainability was first evaluated with the C-means clustering algorithm, and the results were subsequently incorporated into a stochastic mathematical model for network design. Similarly, [Sabouhi et al. \(2021\)](#) employed K-means clustering to group regions based on sustainability performance for designing a sustainable and resilient SCND. Later, [Gholami-Zanjani et al. \(2021\)](#) explored resilient SCND in the context of epidemic disruptions, analyzing cascading effects and using fuzzy c-means clustering to support scenario reduction. Recently, [Jafarian et al. \(2025\)](#) addressed pharmaceutical SCND by considering resilience, sustainability and digitalization. Their approach evaluated potential suppliers with a random forest regressor and developed a two-stage stochastic model for network design. A comparable methodology was applied by [Zeynali et al. \(2025\)](#) in the agri-food industry. [Table 1](#) summarizes and compares the most relevant studies.

Based on the literature review and the analysis in [Table 1](#), it is evident that prior studies have overlooked the role of disruption inter-arrival times. Furthermore, the integration of ML techniques into resilient SCND remains limited, with only a few publications addressing this intersection. Despite the acknowledged importance of the ripple effect, especially under super-disruptions such as pandemics, it has not yet been incorporated into existing SCND models. Additionally, the use of hyper-heuristics as advanced solution methods has not been explored in this context. In light of these gaps, this study makes the following key contributions.

- o Proposing a novel hybrid methodology that integrates machine learning, a phase-type disruption model, and a stochastic optimization framework for resilient SCND.

Table 1

Key attributes of the selected studies.

Reference	Supply chain network design	Supply chain characteristic	Network structure	Impact of inter-arrival time of disruptions	Applying ML methods	Considering ripple effect	Uncertainty approach	Solution method			
								Approximation			
								LR	Mh	HH	Heu
Azad et al. (2013)	✓	Rs	F			–		BD, OS			
Salehi Sadghiani et al. (2015)	✓	Rs	F			Rb		OS			
Hasani and Khosrojerdi (2016)	✓	Rs	F			Rb			✓		
Jabbarzadeh et al. (2016)	✓	Rs	F			Rb		OS	✓		
Fattahi et al. (2017)	✓	Rs, Rp	F			St		OS			
Jabbarzadeh et al. (2018)	✓	Rs, Su	F		✓	St		OS			
Sabouhi et al. (2018)	✓	Rs	F			St		OS			
Sabouhi et al. (2020)	✓	Rs	F			St		BD, OS, Oth			
Gholami-Zanjani et al. (2021)	✓	Rs	F		✓	St		BD, OS			
Hasani et al. (2021)	✓	Rs, Su	F			St, Rb			✓		
Sabouhi et al. (2021)	✓	Rs, Su	F		✓	St, Rb		BD, OS			
Sawik (2022)		Rs	F			✓	St	OS			
Vali-Siar and Roghanian (2022)	✓	Rs, Su, Rp	OL, CL			St, Rb	OS		✓		
Vali-Siar et al. (2022)	✓	Rs, Oth	OL, CL			St	OS		✓		
Foroozesh et al. (2023)	✓	Rs, Su	F			Fz, Rb	OS				
Sindhwanvi et al. (2023)		Rs, Oth	F			✓	St	OS	✓		
Sawik (2023)		Rs, Vb	F			✓	St	OS			
Alikhani et al. (2023)	✓	Rs	F			St		OS			
Tsao et al (2024)	✓	Rs, Su	F			Fz	OS, Oth				
Jafarian et al. (2025)	✓	Rs, Su	F		✓	St	OS, Oth				
Nili et al. (2025)	✓	Rs, Su	CL			St	OS				
Yilmaz et al. (2025)	✓	Rs, Rp, Vb	F			✓	St, Oth		✓		
Zeynali et al. (2025)		Rs, Su, Oth	F, R		✓		Fz, Oth				
Zheng et al. (2025)	✓	Rs	F			St, Fz	Oth				
This paper	✓	Rs	F	✓	✓	✓	St	OS	✓	✓	✓

Resilience (Rs), Sustainability (Su) Responsiveness (Rp), Viability (Vb) Other (Oth); Forward (F), Reverse (R), Open-loop (OL), Closed-loop (CL); Stochastic programming (St), Robust optimization (Rb), Fuzzy programming (Fz), Other (Oth); Optimization software (OS), Lagrangian relaxation/ decomposition (LR), Benders decomposing (BD), Metaheuristics (Mh), Hyper-heuristics (HH), Heuristics (Heu).

- o Considering disruption inter-arrival times, super-disruption impacts, and the ripple effect.
- o Developing a new two-stage stochastic model for the resilient SCND problem.
- o Presenting a new hyper-matheuristic algorithm to address computational complexity in large-scale instances.
- o Validating the proposed approach using an industrial case study from the agricultural supply chain.

3. Problem description

In this study, we address the design of a resilient SC network that is exposed to the risk of super-disruptions and disruptions. Our aim is to model and optimize a SC that is able to withstand and recover from such disruptions by adopting a set of practical resilience strategies.

The SC considered is structured as a four-echelon network as depicted in Fig. 1. Raw materials and parts are supplied by the suppliers. The first-stage production centers process raw materials and parts, and send the products to the second-stage production centers, where the final processing is completed. Finished products are delivered to customers either directly or via DCs.

In the considered supply chain network, disruptions may arise from various sources. They can occur directly at a facility as ordinary disruptions, as super-disruptions, or indirectly through the ripple effect propagated from super-disruptions affecting other parts of the network. Regardless of their origin, these disruptions can partially or fully reduce the operational capacity of production centers and distribution centers. The model is designed to explicitly capture the disruptions using some scenarios. Further details on the specific disruption mechanisms and the effects of their inter-arrival times and the approach for generate scenarios based on the available data are provided in the subsequent subsections. The main problem addressed is to design a resilient supply chain network capable of effectively absorbing and recovering from disruptions by applying a number of effective resilience strategies. These include the proactive measure of multiple sourcing, as well as the reactive strategies of dual-channel distribution, capacity expansion, outsourcing, and facility fortification. Among these, multiple sourcing is the most widely recognized approach for managing disruptions (Jabbarzadeh et al., 2018), as it enables facilities to receive inputs from two or more upstream suppliers. This is particularly advantageous when one or more facilities experience shutdowns or capacity reductions due to disruptions. In facility fortification, a specific level of fortification is chosen for each facility, with higher levels resulting in a reduced negative impact from disruptions but at a higher cost (Fattah et al., 2017). Capacity expansion involves increasing facility capacity in response to disruptions to partially offset the reduction; however, this initiative incurs additional costs and is limited to a predetermined maximum (Vali-Siar and Roghanian, 2022). The dual-channel distribution strategy allows products to reach customers both directly from factories and through DCs, which is particularly valuable when DCs have diminished capacity or are non-operational. Outsourcing entails sourcing products from external facilities outside the SC network to maintain flow, though it is assumed to incur higher costs compared to internal production. The main assumptions of the problem are as follows.

- o Suppliers, production centers, and DCs all operate with bounded capacities, which may be partially or fully reduced under disruption scenarios.
- o Due to the uncertain effects of disruptions, the percentage of capacity loss at each facility due to disruptions is treated as a stochastic parameter.
- o Customer locations and demand levels are assumed to be predetermined and constant throughout the planning horizon.
- o The model considers multiple raw materials products.
- o All flows and decisions are planned for a single planning horizon.

3.1. Mathematical modeling

A MILP formulation is developed for the investigated SCND problem. The scenario-based two-stage stochastic programming approach is used to handle the aforementioned uncertainties. In two-stage stochastic programming, the decision variables are divided

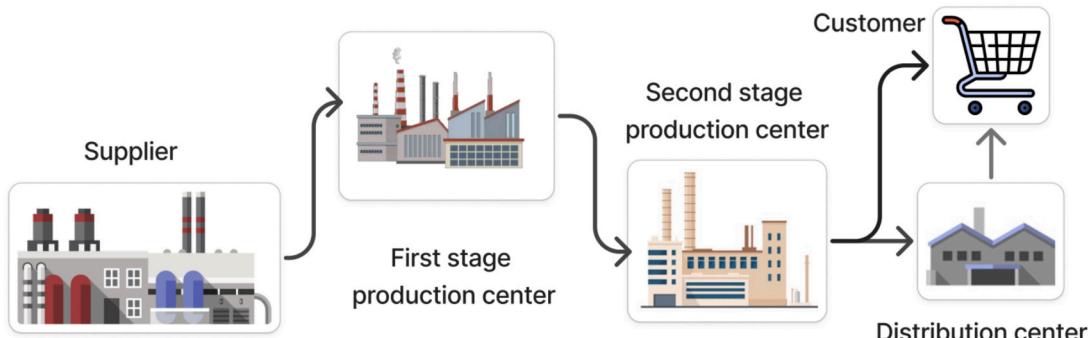


Fig. 1. An illustration of the described supply chain network.

into two categories: first-stage variables and second-stage variables. First-stage decisions are made independently of the scenarios. Second-stage decisions are made after one of the possible scenarios occurs and therefore depend on the defined scenarios. The general form of a scenario-based two-stage stochastic mathematical model is as follows:

$$\text{Min}Z = c^T x + E_s \varphi(x, s)$$

$$E_s \varphi(x, s) = \sum_{s \in S} \pi_s b_s^T y_s$$

$$Ax = D$$

$$B_s x + D_s y_s = e_s \quad \forall s \in S$$

$$x \geq 0$$

$$y_s \geq 0 \quad \forall s \in S,$$

where B_s , D_s , b_s and e_s are stochastic parameters. $S = \{1, 2, \dots, N\}$ is the set of scenarios and π_s is the probability of scenario s . The objective function consists of two parts, the cost of first-stage variables ($c^T x$) and the average of the cost of second-stage variables over scenarios, $E_s \varphi(x, s)$.

Sets, parameters, and variables are defined as follows. Based on the description provided in the problem statement, we define the sets required for the mathematical model. As shown, each echelon of the supply chain is represented by a corresponding set because these elements are explicitly needed in the formulation (see Fig. 1). In addition, a set of disruption scenarios is introduced to incorporate uncertainty within the scenario-based stochastic programming framework (Table 2, Table 3 and Table 4).

A set of parameters is introduced to define the key input data for the mathematical model. These parameters specify the network's physical and operational characteristics, including facility capacities, material conversion ratios, costs, product prices, and uncertainty factors related to disruptions.

The developed mathematical model determines the selection of suppliers and the locations of factories and distribution centers as first-stage decision variables. Another crucial aspect of network design involves determining the flow quantities of materials and products between supply chain echelons. These values are established as second-stage decision variables by the model. The decision variables are as follows.

The detailed mathematical model is presented in the following.

Objective function

The objective function (1) is to maximize the total profit of the SC by aggregating revenues and subtracting all cost components, namely fixed facility establishment costs, raw material procurement costs, production and processing costs, capacity-expansion costs, outsourcing costs, and transportation costs.

$$\begin{aligned} \text{Max } Z = & \sum_s \pi_s \sum_r \sum_c pr_r \left(\sum_{p'} (qr_{rp'cs}) + \sum_k (qc_{rkcs} + qo_{rkcs}) \right) \\ & - \left(\sum_{f \in \{p, p', k\}} fe_f x_f + \sum_i fs_i ss_i \right. \\ & + \sum_s \pi_s \left(\sum_r \sum_i \sum_p cr_i qm_{rips} + \sum_r \sum_p \sum_p mc_p qp_{rpp's} + \sum_r \sum_{p'} \sum_c mc_{p'} qr_{rp'cs} \right. \\ & + \sum_r \sum_{p'} \sum_k mc_{p'} qd_{rp'ks} + \sum_r \sum_k \sum_c cd_k qc_{rkcs} + \sum_p ec_p qe_{ps} + \sum_{p'} ec_{p'} qe_{p's} \\ & + \sum_r \sum_k \sum_c co_r qo_{rkcs} + \sum_r \sum_i \sum_p tc_{ip} qm_{rips} + \sum_r \sum_p \sum_{p'} tc_{pp'} qp_{rpp's} \\ & \left. \left. + \sum_r \sum_{p'} \sum_k tc_{p'k} qd_{rp'ks} + \sum_r \sum_{p'} \sum_c tc_{p'c} qr_{rp'cs} + \sum_r \sum_k \sum_c tc_{kc} (qc_{rkcs} + qo_{rkcs}) \right) \right) \end{aligned} \quad (1)$$

Table 2
List of sets.

Sets:	
i	Index of suppliers, $i \in I$
r	Index of raw material and final product types, $r \in R$
p	Index of first stage production centers, $p \in P$
p'	Index of second stage production centers, $p' \in P'$
j	Index of production centers, $\{p, p'\} \in J$
k	Index of DCs, $k \in K$
f	Index of facilities and customers, $\{i, p, p', k\} \in F$
c	Index of customers, $c \in C$
s	Index of scenarios, $s \in S$

Table 3

List of parameters.

Parameters:	
Cost parameters	
f_{si}	Cost of choosing supplier i
fe_f	Cost of establishing facility f ($f \in \{p, p', k\}$)
$tc_{ff'}$	Unit shipment cost between facilities f and f' ($f, f' \in \{i, p, p', k, c\}$)
mc_j	Unit manufacturing cost in production center j
ec_j	Unit cost of adding extra manufacturing capacity in production center j
cd_k	Unit cost of processing product in DC k
cr_i	Unit cost of purchasing raw material type r from supplier i
co_r	Unit cost of outsourcing product type r
Processing yield coefficients	
δ	Conversion factor of raw materials to product
γ	Conversion ratio at the second-stage production centers, representing the fraction of final product obtained from processing one unit of input product
d_c	Demand of customer c for product type r
Capacity parameters	
cap_j	Capacity of production center j
cap_{ri}	Capacity of supplier i for material type r
$cad_{p'}$	Direct distribution capacity of production center p'
Mor	Maximum amount of products type r that can be supplied by outsourcing
ce_j	Maximum extendable capacity in production center j
Revenue parameter	
pr_r	Price of product type r
Disruption and scenario parameters	
λ_{fs}	The fraction of lost capacity of facility f ($f \in \{i, p, p', d\}$) under scenario s
π_s	Probability occurrence of scenario s

Table 4

List of variables.

Decision variables:	
Design/location variables (first stage)	
ss_i	1, if supplier i is chosen, 0 otherwise
x_f	1, if facility f ($f \in \{p, p', k\}$) is established, 0 otherwise
Material flow variables (second stage variables)	
qm_{ips}	Amount of raw material type r shipped from supplier i to production center p under scenario s
$qp_{rpp'}$	Amount of product type r transported from production center p to production center p' under scenario s
$qr_{p'cs}$	Amount of final product type r transported from production center p' to customer c under scenario s
$qd_{rp'ks}$	Amount of final product type r transported from production center p' to DC k under scenario s
qc_{rkcs}	Amount of final product type r transported from DC k to customer c under scenario s
qo_{rcks}	Amount of product type r supplied by outsourcing transported from DC k to customer c under scenario s
Capacity expansion variable (second stage variable)	
qe_js	Amount of added capacity to production center j to under scenario s

Subject to:

Material Flow Balance Constraints

These constraints maintain flow conservation throughout the network, ensuring that the quantity of materials and products entering each node (production center, DC, or customer) is balanced with the quantity leaving, in accordance with the production and distribution structure. Constraints (2) limit first-stage production by the available raw material input multiplied by its conversion factor. Constraints (3) enforce material balance at each second-stage facility by equating output (to distribution centers and direct shipments) with the converted input. Constraints (4) ensure the flow conservation at each distribution center by equating inbound and outbound quantities.

$$\sum_{p'} qp_{rpp's} = \delta \sum_i qm_{rips}, \forall r, p, s \quad (2)$$

$$\sum_c qr_{rp'cs} + \sum_k qd_{rp'ks} = \gamma \sum_p qp_{rpp's}, \forall r, p', s \quad (3)$$

$$\sum_{p'} qd_{rp'ks} = \sum_c qc_{rkcs}, \forall r, k, s \quad (4)$$

Demand Satisfaction Constraints

Constraints (5) allow full or partial fulfillment of customer demand.

$$\sum_{p'} qr_{rp'cs} + \sum_k qc_{rkcs} + \sum_k qo_{rkcs} \leq d_{rc}, \forall r, c, s \quad (5)$$

Capacity Constraints

Constraints (6)–(10) ensure that production and distribution at each facility do not exceed their respective capacities under each disruption scenario, taking into account reductions caused by disruptions as well as any added capacity from expansion strategies.

$$\sum_p qm_{rips} \leq (1 - \lambda_{is}) cap_{riSS_i}, \forall i, r, s \quad (6)$$

$$\sum_r \sum_{p'} qp_{rpp's} \leq ((1 - \lambda_{ps}) cap_p) x_p + qe_{ps}, \forall p, s \quad (7)$$

$$\sum_r \sum_c qr_{rp'cs} + \sum_r \sum_k qd_{rp'ks} \leq ((1 - \lambda_{p's}) cap_p) x_{p'} + qe_{p's}, \forall p', s \quad (8)$$

$$\sum_r \sum_c qc_{rkcs} \leq cad_p x_{p'}, \forall p', s \quad (9)$$

$$\sum_r \sum_c qo_{rkcs} + \sum_r \sum_k qo_{rkcs} \leq ((1 - \lambda_{ks}) cap_k) x_k, \forall k, s \quad (10)$$

Capacity expansion and outsourcing Constraints

Constraints (11) and (12) specify the allowable use and limits of resilience strategies such as capacity expansion, while Constraints (13) define the allowable outsourcing at each facility.

$$qe_{ps} \leq ce_p x_p, \forall p, s \quad (11)$$

$$qe_{p's} \leq ce_p x_{p'}, \forall p', s \quad (12)$$

$$\sum_r \sum_k qo_{rkcs} \leq Mo_r, \forall r, s \quad (13)$$

Domain Constraints

Constraints (14) define the domain of all binary, integer, and continuous decision variables.

$$\begin{aligned} qm_{rips}, qp_{rpp's}, qr_{rp'cs}, qc_{rkcs}, qd_{rp'ks}, qo_{rkcs}, qe_{js} &\geq 0, \\ ss_i, x_f &\in \{0, 1\}. \end{aligned} \quad (14)$$

4. Discussion on disruptions and the proposed hybrid approach

We now examine the impact of inter-arrival times between disruptions on the resulting damages, specifically the reduction in facility capacity. Consequently, a two-phase hybrid approach is developed for designing the described SC network. In Phase 1, a clustering algorithm is first applied to detect the dependency structure between inter-arrival times and damages based on disruption data. Next, PH distributions and the related theorems are used to determine the probability distributions of cumulative damages. These probability distributions are then utilized to generate scenarios that incorporate disruption risks into the optimization model. In Phase 2, the proposed mathematical model is solved to design the SC network and determine the optimal values of the decision variables.

Algorithm 1 outlines the steps of the developed hybrid approach.

Algorithm 1. Proposed two-phase hybrid approach.

Phase 1	Machine learning and PH distributions
Stage 1.	Collecting data of disruptions (times and damages)
Stage 2.	Applying the genetic clustering algorithm and obtain clusters of disruptions based on their damages.
Stage 3.	Using PH distributions and the related theorems to obtain probability distributions of damages
Stage 4.	Generating disruption scenarios using Monte Carlo simulation based on the found distributions and implementing scenario reduction
Phase 2	Mathematical optimization
Stage 1.	Developing the stochastic MILP model using the disruption scenarios
Stage 2.	Solving the mathematical model using the proposed solution methods

As noted in the introduction, when a super-disruption occurs in a particular region (e.g., a pandemic), it often propagates throughout the entire supply chain, subsequently affecting other regions. This phenomenon is known as the ripple effect. The proposed approach inherently incorporates this effect. Specifically, the historical data used to fit the phase-type distributions include both primary disruptions and those propagated from other facilities (ripple-induced disruptions). Consequently, the estimated probability distributions and the generated scenarios naturally capture the ripple effect. As previously mentioned, the ripple effect itself is also effective in reducing the inter-arrival time between disruptions and its impacts on the severity of disruptions is considered by the proposed method. The detailed modeling approach and the underlying theoretical foundations are presented in the following subsections.

In this research, the scenario-based stochastic programming approach is employed to include disruptions in the mathematical model. For this purpose and to generate scenarios, one should be aware of probability distributions of disruptions' damages. In this article, disruption scenarios are discussed in more depth. Scenario generation is the discretization procedure of the continuous probability density function of stochastic parameters through a set of discrete scenarios (Fattahi et al., 2017). For the described problem setting, the occurrence of disruptions follows a Bernoulli distribution (Mak and Shen, 2012). Consider a SC whose facilities are under disruptions. To formulate the problem, it is assumed that the probability of the occurrence of each disruptive event or shock follows a Bernoulli distribution with parameter p_{fac} ($p_{fac} \in [0, 1]$), which denotes the parameter of Bernoulli distribution related to the occurrence of disruptive events in facility fac , ($fac \in \{i, p, j, h, f\}$). Based on this, m_{ds}^{fac} is defined, which equals 1 if disruptive event d ($d \in \{d_1, d_2, \dots, d_D\}$) occurs for facility fac under scenario s . Then, we define the parameter M_s^{fac} which equals 1, if at least one disruptive event occurs under scenario s ; This means that: $M_s^{fac} = 1 | \sum_d m_{ds}^{fac} \geq 1 \forall fac, s$. Parameter ψ_s^{fac} represents the disrupted fraction of capacity of facility fac under scenario s . When disruptions occur during the planning horizon, involving one or more events or shocks, the facility's capacity decreases partially or completely. The challenge is in computing ψ_s^{fac} caused by one or more disruptive events. This is because, in reality, disruptive events may follow different probability distributions. Also, a facility may be disrupted itself or may be disrupted by the ripple effect.

For scenario generation, a general framework is considered in which there are no specific assumptions on the distributions of the inter-arrival times of disruptions and their damage magnitudes. In real-world applications, these variables may follow any distributions. Furthermore, the dependency between the damages of disruptions and their inter-arrival times is also taken into account. If classical distributions such as normal, Weibull, or others are employed for modeling this problem, we face challenges, including:

- o Identifying and fitting an appropriate well-known distribution to the data can be difficult, and in many cases, none of these distributions provide a sufficiently accurate fit. The inter-arrival time between two consecutive disruptions and their damage magnitudes may follow complex distributions due to system variability.
- o Computations of convolutions are required for the variables; however, these convolutions are difficult to determine except for some distributions such as exponential, normal, and gamma.
- o To handle dependencies, the data are divided into clusters. However, computing the convex mixture sum of their distributions is required, which is difficult to determine with well-known distributions.

To address these challenges, the use of pH distributions and their properties are proposed for the following reasons (Bladt and Nielsen, 2017):

- o Distribution of each variable with positive support can be arbitrarily approximated by a PH distribution.
- o PH distributions are closed under finite convolutions operation. The sum of two or more independent PH-distributed random variables is itself PH-distributed. Hence, modeling cumulative damages and aggregated inter-arrival times of disruptions remain analytically tractable.
- o PH distributions are closed under finite mixtures operation. Convex mixture sum of two or more independent PH distributions also results in another PH distribution. This property makes PH distributions especially useful for representing PH distribution of clustered damage data.

4.1. Phase type distribution

The *PH* distributions are developed based on the concept of states in a Markov chain in which each state represents a phase (Riascos-Ochoa et al., 2014). Assume a continuous-time Markov Chain whose state space $S = \{1, 2, \dots, n, n+1\}$, where the first n states are transient, the state $n+1$ is absorbing and the generator matrix is as follows.

$$\mathbf{Q} = \begin{bmatrix} \mathbf{D} & \mathbf{d} \\ 0 & 0 \end{bmatrix}, \quad (15)$$

where \mathbf{D} is an $n \times n$ matrix and shows the transition intensities between transient states and the column vector \mathbf{d} shows the transition intensities between transient states and absorbing states. Let $\mathbf{d} = -\mathbf{De}$, where \mathbf{e} is a column vector of ones. The initial probability vector is denoted with (ρ, ρ_{n+1}) . $\rho = (1, 2, \dots, n)$ includes the initial probabilities of transient states. ρ_{n+1} denotes the initial probability of absorbing state $n+1$. Let Z be a random variable denoting the time to reach the absorbing state $n+1$, then Z has a *PH* distribution with parameters (ρ, \mathbf{D}) and order n (the order is the number of transient states). Based on the descriptions, the probability density function and the cumulative density function of the variable Z are as follows:

$$f(z) = \rho \cdot \exp(\mathbf{D}z) \cdot \mathbf{d}; z \geq 0 \quad (16)$$

$$F(z) = 1 - \rho \cdot \exp(\mathbf{D}z) \cdot \mathbf{e}; z \geq 0. \quad (17)$$

Various algorithms can be applied for fitting a *PH* distribution to non-negative data. These algorithms can be divided into two main groups, including moment matching (MM) algorithms and expectation maximization (EM) algorithms (Verbeelen, 2013; Riascos-Ochoa et al., 2014). According to Verbeelen (2013) and Bladt and Nielsen (2017), two theorems on *PH* distributions that are applied in our study can be expressed as follows.

Theorem 1. (Let X $PH(\alpha, A)$ and Y $PH(\beta, B)$. Then the convex mixture sum $Z = p_1X + p_2Y$ where $p_1 + p_2 = 1$ follows *PH* distribution with representation (ρ, P) as follows:)

$$\rho = [p_1\alpha, p_2\beta] \text{ and } P = \begin{bmatrix} A & 0 \\ 0 & B \end{bmatrix} \quad (18)$$

Theorem 2. (Let X $PH(\alpha, A)$ and Y $PH(\beta, B)$ of orders n_X and n_Y respectively. Then $X + Y$ $PH(\rho, P)$ follows *PH* distribution with representation (ρ, P) as follows:)

$$\rho = [\alpha, \alpha_{n_X+1}\beta] \text{ and } P = \begin{bmatrix} A & \alpha\beta \\ 0 & B \end{bmatrix} \quad (19)$$

In Eq. (19), the α_{n_X+1} represents the initial probability of absorbing state n_X+1 and $\alpha = -Ae$.

4.2. A presentation of the *PH*-disruption model

Let the inter-arrival time between two consecutive disruptive events $\{T_j\}_{j \geq 1}$ follow *PH* distributed with $PH(\gamma_j, T_j)$ of order n_{T_j} such that $t_j = -T_j e$. The damages caused by disruptive events are members of cluster r with the below probability:

$$pc_i = \begin{cases} P(T_j \leq \Delta_i) = 1 - \gamma_j \cdot \exp(-T_j \Delta_i) \cdot e; & i = 1, \\ P(\Delta_{i-1} < T_j \leq \Delta_i) = \gamma_j \cdot \exp(-T_j \Delta_{i-1}) \cdot e - \gamma_j \cdot \exp(-T_j \Delta_i) \cdot e; & 1 < i < r, \\ P(\Delta_{i-1} < T_j) = \gamma_j \cdot \exp(-T_j \Delta_{i-1}) \cdot e; & i = r. \end{cases} \quad (20)$$

Therefore, each disruptive event with probability pc_r is a member of r^{th} cluster that has damage W_{rj} . Let W_{rj} follow *PH* distribution with $PH(\omega_{rj}, W_{rj})$ of order $n_{W_{rj}}$. Based on Theorem 1, the damage corresponding to each disruptive event follows *PH* distribution as below.

W_j $PH(\omega_j, W_j)$, where $\omega_j = [pc_1\omega_{1j}, pc_2\omega_{2j}, \dots, pc_{r-1}\omega_{r-1j}, pc_r\omega_{rj}]$,

$$W_j = \begin{bmatrix} W_{1j} & 0 & 0 \\ 0 & W_{2j} & 0 & 0 \\ \ddots & \ddots & \ddots & \ddots \\ \ddots & \ddots & W_{k=r-1j} & 0 \\ 0 & & & W_{rj} \end{bmatrix} \quad \text{and} \quad w_j = -W_j e. \quad (21)$$

According to Theorem 2, the total damage caused by n disruptive events, $D_n = \sum_{j=1}^n W_j$, follows $PH(\pi_n, D_n)$ distribution (Shamstabar et al., 2021):

$$\boldsymbol{\pi}_n = [\omega_1, 0, \dots, 0],$$

$$\mathbf{D}_n = \begin{bmatrix} \mathbf{W}_1 & \mathbf{w}_1 \cdot \omega_2 & 0 \\ 0 & \mathbf{W}_2 & \mathbf{w}_2 \cdot \omega_3 & 0 \\ & \ddots & \ddots & \ddots & \ddots \\ & & \ddots & \mathbf{W}_{n-1} & \mathbf{w}_{n-1} \cdot \omega_n \\ & & & 0 & \mathbf{W}_n \end{bmatrix} \quad (22)$$

Also, according to [Theorem 2](#), the total time after n disruptive events, $S_n = \sum_{j=1}^n T_j$, has PH distribution of order $n_{T_1} + n_{T_2} + \dots + n_{T_n}$, where can be represented as below ([Shamstabar et al., 2021](#)):

$$\boldsymbol{\beta}_k = [\gamma_1, 0, \dots, 0],$$

$$\mathbf{S}_k = \begin{bmatrix} T_1 & t_1 \cdot \gamma_2 & 0 \\ 0 & T_2 & t_2 \cdot \gamma_3 & 0 \\ & \ddots & \ddots & \ddots & \ddots \\ & & \ddots & T_{n-1} & t_{n-1} \cdot \gamma_n \\ & & & 0 & T_n \end{bmatrix} \quad (23)$$

Then, the probability of occurrence of k disruptive events until time t in is ([Nakagawa, 2007](#)):

$$P\{S_k \leq t\} = F^{(n)}(t) = \begin{cases} 1 - \beta_n \cdot \exp(S_n t) \cdot e, & n \geq 1 \\ 1, & n = 0 \end{cases} \quad (24)$$

The cumulative damage depends on the number of disruptions in the planning period. Therefore, the cumulative damage value should be approximated. To this end, for a given approximation error $\varepsilon > 0$ the computation is truncated at the N th disruptive event such that $F^{(k)}(t) < \varepsilon$. Using [Theorem 2](#), the approximate distribution of damages (DP) can be derived as follows.

DP PH($\boldsymbol{\pi}_N, \mathbf{D}_N$), where:

$$\boldsymbol{\pi}_N = [\omega_1, 0, \dots, 0] \text{ and} \\ \mathbf{D}_N = \begin{bmatrix} \mathbf{W}_1 & \mathbf{w}_1 \cdot \omega_2 & 0 \\ 0 & \mathbf{W}_2 & \mathbf{w}_2 \cdot \omega_3 & 0 \\ & \ddots & \ddots & \ddots & \ddots \\ & & \ddots & \mathbf{W}_{N-1} & \mathbf{w}_{N-1} \cdot \omega_N \\ & & & 0 & \mathbf{W}_N \end{bmatrix} \quad (25)$$

4.3. Algorithm for clustering the damages of disruptions

ML involves methods for finding patterns in data and using them to forecast future data and make decisions under uncertain conditions ([Murphy, 2012](#)). ML has been utilized in various fields, including transportation, health, and finance, and there are a few areas in which this concept has not been used ([Xu and Saleh, 2021](#)). It is divided into three main groups of supervised learning, unsupervised learning, and semi-supervised learning. The unsupervised learning method, which has been used in this paper, places data in clusters that have similar characteristics so that the clusters and their number are not known in advance, and, in fact, the allocation of data to meaningful clusters is the output of this approach ([Xu and Saleh, 2021](#)).

Assume that there is a data set $Q = \{(T_1, W_1), (T_2, W_2), \dots, (T_L, W_L)\}$ constituted from L order pairs and let $X = \{W_1, W_2, \dots, W_L\}$. The mission of the clustering method is to position set X in r subsets or clusters, C_1, \dots, C_r . Equation (26) represents the performance measure (PM) of the clustering. This measure is, in fact, the sum of the squared Euclidean distance between each data point W_l and centroid m_k of the cluster C_r which includes W_l .

$$PM = \sum_{l=1}^L \sum_{r=1}^M I(W_l \in C_r) \|W_l - m_r\|^2, \quad (26)$$

where $I(W_l \in C_r)$ is 1, if W_l belongs to cluster C_r and 0 otherwise.

The data are clustered with $r-1$ separator lines so that the performance measure has its minimum possible value. Classic clustering methods often converge to a local optimal solution, which may result in inaccurate clustering like k-means. Moreover, clustering algorithms utilizing dynamic programming and branch and bound techniques are only suitable for small datasets. Also, usually exact algorithms for clustering are time-consuming. Metaheuristic algorithms are recognized as effective optimization techniques for achieving global or near-global solutions. They are capable of handling variables of both discrete and continuous nature and have found extensive use in solving clustering problems. Some metaheuristics, such as artificial bee colony optimization, particle swarm optimization, and ant colony optimization are frequently applied in clustering ([Bagirov et al., 2020](#)).

We utilized the genetic-based clustering algorithm proposed by [Shamstabari et al. \(2024\)](#). In this procedure, the set Q is first clustered using generated separator lines (solutions), the quality of this clustering is then assessed using Equation (42). For more details interested readers are referred to the paper of [Shamstabari et al. \(2024\)](#).

4.4. The generation of disruption scenarios and reduction

Disruption scenarios are generated using Monte Carlo simulation based on the fitted phase-type distributions for each facility's capacity loss. For each scenario, and for every facility f , a uniformly distributed random number $r \in [0, 1]$ is first generated. The inverse cumulative distribution function F_f^{-1} of the fitted PH distribution is then applied to obtain a random draw representing the capacity loss lc_f for each facility, i.e., $lc_f = F_f^{-1}(r)$. The losses of all facilities are combined to form a scenario. This process is repeated until the desired number O of initial scenarios, each with equal probability, is obtained.

To decrease the computational complexity, scenario reduction methods are usually applied. These methods work in such a way that the reduced number of scenarios can be a good representative of the original scenario set ([Gholami-Zanjani et al., 2021](#)). In this paper, the well-known fuzzy c-means clustering method (FCM) developed by [Bezdek et al. \(1984\)](#) is used. Using the FCM algorithm, the scenarios are placed in some clusters, and then the center of a cluster serves as the representative of the scenarios of that cluster and will be used in the problem. In other words, the scenarios of each cluster are replaced by the center of that cluster. The basic difference in scenario reduction with FCM in comparison with traditional clustering methods, is that in FCM each scenario is a member of every cluster with a membership degree, while in traditional clustering methods each scenario only belongs to one cluster. Note that the sum of the values of the membership function for each scenario is equal to one. The FCM algorithm is as follows.

i) Fix the number of clusters (S) the weighting exponent (m) and generate an initial membership matrix determining the membership degree of each scenario to each cluster. Set $b = 0$.

$$U^0 = \begin{bmatrix} u_{11} & \dots & u_{1O} \\ \ddots & u_{so} & \ddots \\ u_{s1} & \dots & u_{SO} \end{bmatrix} \quad (27)$$

u_{so} is the membership degree of scenario o in cluster s ; $0 \leq u_{so} \leq 1 \sum_{s=1}^S u_{so} = 1$.

ii) Compute the center vectors as follows.

$$C_s = \frac{\sum_{o=1}^O u_{so}^m \xi_o}{\sum_{o=1}^O u_{so}^m}, 1 \leq s \leq S \quad (28)$$

ξ_o is the main characteristic of scenario o (Here, the amount of disrupted capacity (damage)).

iii) Compute the updated membership matrix as follows.

$$U_{so}^{b+1} = \left(\sum_{j=1}^S \left(\frac{d_{so}}{d_{js}} \right)^{\frac{2}{m-1}} \right)^{-1} \quad (29)$$

In the above equation $d_{so} = \|\xi_o - C_s\|_A$ ($\|\cdot\|_A$ is induced A-norm on \mathbb{R}^n)

$$U^{b+1} = \begin{bmatrix} u_{11}^{b+1} & \dots & u_{1O}^{b+1} \\ \ddots & u_{so}^{b+1} & \ddots \\ u_{s1}^{b+1} & \dots & u_{SO}^{b+1} \end{bmatrix} \quad (30)$$

iv) If $\|U^{b+1} - U^b\| < \epsilon$ stop. Otherwise, set $b = b + 1$ and go to step 2.

We used 2-norm for the computations. After the algorithm terminates, $\pi_s = \sum_{o=1}^O u_{so} p_o$ is computed as the probability of each scenario. As clarified, the scenarios are generated using existing data and the identified distribution, resulting in a data-driven scenario-based stochastic programming (DDSSP) approach.

5. Solution methodology

SCND problems are classified as NP-hard ([Govindan et al., 2016](#)). In order to deal with the complexity of the problem and obtain high-quality solutions in problems with large sizes, a Lagrangian decomposition method and a novel hyper-matheuristic are proposed.

5.1. Lagrangian decomposition method: obtaining upper bound

Lagrangian relaxation has been proven to be an efficient solution method for solving MILP models like SCND problems ([Vali-Siar and Roghanian, 2022](#)). In this paper, we utilize Lagrangian decomposition to obtain a suitable upper bound and compare these bounds with the results of other proposed solution methods. In this method, the upper bound (or lower bound in minimization problems) can be improved during an iterative procedure using the Lagrangian dual model. In order to construct the Lagrangian dual problem, constraints (3) and (8) are relaxed and added to the objective function using multipliers $\alpha_{rp's}$ and $\beta_{p's}$. By doing so, the following model

is obtained:

$$\begin{aligned}
 \text{Max } Z = & \sum_s \pi_s \sum_r \sum_{p'} \sum_c pr_r \left(qr_{rp'cs} + \sum_k (qc_{rkcs} + qo_{rkcs}) \right) \\
 & - \left(\sum_{f \in \{p, p'\}} fe_f x_f + \sum_i fs_i ss_i \right. \\
 & + \sum_s \pi_s \sum_r \sum_i \sum_p cr_{ri} qm_{rips} + \sum_r \sum_p \sum_{p'} mc_p qp_{rpp's} + \sum_r \sum_{p'} \sum_c mc_{p'} qr_{rp'cs} \\
 & + \sum_r \sum_{p'} \sum_k mc_p qd_{rp'ks} + \sum_r \sum_k \sum_c cd_k qc_{rkcs} + \sum_p ec_p qe_ps + \sum_{p'} ec_{p'} qe_{p's} \\
 & + \sum_r \sum_k \sum_c co_r qo_{rkcs} + \sum_r \sum_i \sum_p tc_{ip} qm_{rips} + \sum_r \sum_p \sum_{p'} tc_{pp'} qp_{rpp's} \\
 & + \sum_r \sum_{p'} \sum_k tc_{p'k} qd_{rp'ks} + \sum_r \sum_{p'} \sum_c tc_{p'c} qr_{rp'cs} + \sum_r \sum_k \sum_c tc_{kc} (qc_{rkcs} + qo_{rkcs})) \\
 & + \sum_s \pi_s \left(\sum_r \sum_{p'} \sum_s \alpha_{rp's} \left(\sum_c qr_{rp'cs} + \sum_k qd_{rp'ks} - \gamma \sum_p qp_{rpp's} \right) \right. \\
 & \left. + \sum_{p'} \beta_{p's} \left((1 - \lambda_{p's}) cap_{p'} x_{p'} + qe_{p's} - \sum_r \sum_c qr_{rp'cs} - \sum_r \sum_k qd_{rp'ks} \right) \right)
 \end{aligned} \tag{31}$$

Subject to: Constraint (2), (4)-(7) and (9)-(14).

Closer examination of the above model reveals that the model can be decomposed into two sub-problems. The first problem includes qm_{rips} , $qp_{rpp's}$, qe_{js} , x_p , $x_{p'}$ and ss_i , and the second problem contains $qr_{rp'cs}$, qc_{rkcs} , $qd_{rp'ks}$, qo_{rkcs} and x_k as variables.

The first sub-problem is as follows:

$$\begin{aligned}
 \text{Max } Z = & - \left(\sum_{f \in \{p, p'\}} fe_f x_f + \sum_i fs_i ss_i + \sum_s \pi_s \left(\sum_r \sum_i \sum_p cr_{ri} qm_{rips} + \sum_r \sum_p \sum_{p'} mc_p qp_{rpp's} \right. \right. \\
 & + \sum_p ec_p qe_ps + \sum_{p'} ec_{p'} qe_{p's} + \sum_r \sum_i \sum_p tc_{ip} qm_{rips} + \sum_r \sum_p \sum_{p'} tc_{pp'} qp_{rpp's} \\
 & \left. \left. + \sum_s \pi_s \sum_r \sum_{p'} \alpha_{rp's} \gamma \sum_p qp_{rpp's} + \sum_{p'} \beta_{p's} \left((1 - \lambda_{p's}) cap_{p'} x_{p'} + qe_{p's} \right) \right) \right)
 \end{aligned} \tag{32}$$

Subject to: constraints (2), (6), (7), (11) and (12). Where $\beta_{p's} \geq 0$ and $\alpha_{rp's} \in \mathbb{R}$.

The second sub-problem is as follows:

$$\begin{aligned}
 \text{Max } Z = & \sum_s \pi_s \sum_r \sum_{p'} \sum_c pr_r \left(qr_{rp'cs} + \sum_k (qc_{rkcs} + qo_{rkcs}) \right) - \left(\sum_k fe_k x_k \right. \\
 & + \sum_s \pi_s \left(\sum_r \sum_{p'} \sum_c mc_p qr_{rp'cs} + \sum_r \sum_{p'} \sum_k mc_p qd_{rp'ks} + \sum_r \sum_k \sum_c cd_k qc_{rkcs} \right. \\
 & + \sum_r \sum_k \sum_c co_r qo_{rkcs} + \sum_r \sum_{p'} \sum_k tc_{p'k} qd_{rp'ks} + \sum_r \sum_{p'} \sum_c tc_{p'c} qr_{rp'cs} \\
 & \left. \left. + \sum_r \sum_k \sum_c tc_{kc} (qc_{rkcs} + qo_{rkcs})) \right) + \sum_s \pi_s \left(\sum_r \sum_{p'} \alpha_{rp's} \left(\sum_c qr_{rp'cs} + \sum_k qd_{rp'ks} \right) \right. \\
 & \left. - \sum_{p'} \beta_{p's} \left(\sum_r \sum_c qr_{rp'cs} + \sum_r \sum_k qd_{rp'ks} \right) \right)
 \end{aligned} \tag{33}$$

Subject to: constraints (4), (5), (9), (10) and (13).

Using the subgradient method (Fisher, 2004), the Lagrangian multipliers are updated in each iteration for improving the upper bound using Eqs. (34) and (35).

$$\alpha_{rp's}^{n+1} = \alpha_{rp's}^n + \theta_1^n \left(\sum_c qr_{rp'cs} + \sum_k qd_{rp'ks} - \gamma \sum_p qp_{rpp's} \right) \tag{34}$$

$$\beta_{p's}^{n+1} = \text{Max}\{0, \beta_{p's}^n + \theta_2^n \left(\sum_r \sum_c qr_{rp'cs} + \sum_r \sum_k qd_{rp'ks} - (1 - \lambda_{p's}) cap_{p'} x_{p'} - qe_{p's} \right) \} \tag{35}$$

where the iteration number is denoted by n . θ_1^n and θ_2^n are step size values, which should be computed as follows:

$$\theta_1^n = \frac{\phi(UB^n - LB^n)}{\sum_r \sum_{p'} \sum_c \sum_k (qr_{rp'cs} + qd_{rp'ks} - \gamma qp_{rpp's})^2} \tag{36}$$

Table 5

Description of metaheuristic algorithms used in LLH.

Algorithm	Description
GA	It was initially introduced by Holland (1975) to mimic the process of natural evolution. It starts with an initial population of candidate solutions and applies selection, crossover, and mutation operators iteratively. Selection chooses promising solutions based on fitness; crossover combines two parents to produce offspring; and mutation introduces random changes to maintain diversity.
PSO	It was introduced by Kennedy and Eberhart (1995) and inspired by the social behavior of birds, PSO maintains a population of particles (solutions) that move through the search space by updating their positions based on both personal and global best experiences. Each particle adjusts its velocity using the combination of its own best-known position and that of the entire swarm.
SO	It was developed by Hashim and Hussien (2022) is a bio-inspired algorithm that imitates the hunting behavior of snakes. It dynamically switches between exploration and exploitation modes, enabling adaptive search intensity. Its design encourages diversified movement during the exploration phase and precise adjustments when promising regions are detected.
AO	It was developed by Abualigah et al. (2021) and inspired by the flight strategies of eagles during hunting. It features four search modes—high soar with vertical stoop, contour flight, glide attack, and local walk—each reflecting different degrees of exploration and exploitation.

$$\theta_2^n = \frac{\phi(UB^n - LB^n)}{\sum_r \sum_{p'} \sum_c \sum_k \sum_s (qr_{rp'cs} + qd_{rp'ks} - (1 - \lambda_{ps})cap_{p'}x_{p'} - qe_{ps})^2} \quad (37)$$

where ϕ is a parameter set to 2 in the first iteration and is halved after 10 consecutive iterations provided that no improvement is observed in the upper bound. For computing the lower bound in each iteration, the sub-problems are solved and the values of binary variables are fixed in the original model, and this model is solved subsequently. The Lagrangian decomposition algorithm runs until reaching the stopping criterion which is the optimality tolerance in this study ($(UB - LB)/LB \leq \epsilon$). Note that the obtained upper bound in each iteration is obtained by summing the objectives of two mentioned sub-problems.

5.2. Hyper- matheuristic algorithm

To address the complexity and computational burden of solving SCND, we propose a two-level hyper matheuristic framework that integrates RL with a caching mechanism. The method dynamically selects from a set of metaheuristic operators based on a trained Q-table, while employing a memory-efficient solution list (SL) to avoid redundant evaluations and promote diversity. Moreover, to enhance computational performance and benefit from the accuracy of the exact method, SCND is decomposed into two complementary parts. This decomposition is highly effective because, once the binary variables are fixed, the remaining problem becomes a continuous LP, which can be solved efficiently by GAMS. This allows us to benefit from exact solution accuracy without solving the full MIP at each iteration. The proposed hyper matheuristic framework consists of three layers:

High-Level Strategy (HLS): This layer is responsible for selecting which low-level heuristics to apply at any given point in the optimization process. It does not solve the problem directly but instead learns or decides which low-level method is likely to yield the best progress toward optimality.

Low-Level Heuristics (LLH): These are problem-specific metaheuristics including two well-known metaheuristics: genetic algorithm (GA) and particle swarm optimization (PSO), and two relatively new metaheuristics: Snake optimizer (SO), and Aquila optimizer (AO). Each of these algorithms has its own search dynamics and explores the solution space in different ways. This phase assigns values to binary decision variables. Table 5 provides a brief introduction to the algorithms.

Solving linear programming LP model: At the third layer, the binary variables are fixed and passed to an exact solver (CPLEX solver in GAMS), which solves the resulting linear programming subproblem. In our proposed method, the high-level controller is driven by a reinforcement learning (RL) mechanism, specifically a Q-learning approach, which uses a pre-trained Q-table to guide the heuristic selection. This design allows the algorithm to adapt its strategy over time based on feedback (i.e., improvements in solution quality) received from the low-level heuristics. Algorithm 2 represents the pseudocode of HLS.

Algorithm 2. High-Level RL Hyper-matheuristic main algorithm

- 1 Load pre-trained Q-table
- 2 Set number of episodes (*num_episodes*) for the main loop
- 3 For each episode in *num_episodes*
 - 4 Initialize a random starting solution *current_solution*
 - 5 Calculate its fitness using *objective_function*
 - 6 For each step in fixed number of steps:
 - 7 Select Action using epsilon-greedy:
 - 8 With probability *epsilon*, select a random action (exploration)
 - Otherwise, select the action with the highest Q-value (exploitation)
 - 9 Log the selected action
 - 10 Apply the selected heuristic (GA, PSO, SO, AO):
 - 11 For GA: Run Genetic Algorithm
 - 12 For PSO: Run Particle Swarm Optimization
 - 13 For SO: Run Snake Optimizer
 - 14 For AO: Run Aquila Optimizer
 - 15 Calculate Reward as $reward - fitness_{current} - fitness_{new}$

(continued on next page)

(continued)

Algorithm 2. High-Level RL Hyper-matheuristic main algorithm

```

16   Update Q-table using the formula: $Q_{new}(action\_idx) \leftarrow Q(action\_idx) + \alpha \times (reward + \gamma \times best\_next\_Q - Q(action\_idx))$ 
17   Update current_fitness and current_solution with new values after applying the action.
18   Display the results of the current episode: Best solution and objective value
19   End for
20   Return Best solution and objective value

```

Each episode of the algorithm involves a sequence of decisions where the high-level controller selects and applies one of the LLHs to the current solution. The reward signal is based on the change in objective value, and the Q-table is updated accordingly to reflect the expected utility of each heuristic. During training, rewards are defined as the improvement in solution fitness. To prevent redundant objective function evaluations and reduce reliance the connection with exact solver, we introduce a caching mechanism (Tikani and Setak, 2019). All evaluated solutions are maintained in a sorted list (SL). Upon the generation of a new candidate, its representation is compared against members of SL; if an identical solution is found, the previously computed objective value is retrieved from the cache, thereby obviating a redundant solver invocation. When duplicate solutions occur within the population, only the earliest instance retains its actual fitness, while subsequent copies are assigned an artificially diminished fitness to ensure their removal in the following iteration. Moreover, SL is updated dynamically: any new solution whose performance exceeds that of the current worst-ranked member is inserted at the appropriate position, whereas inferior candidates are discarded. This caching strategy achieves two goals: (i) it prevents unnecessary function evaluations, especially when an external solver is involved, and (ii) it improves diversity by discouraging premature convergence around duplicate solutions. The overall procedure of proposed hyper-matheuristic is summarized in Fig. 2.

6. Computational experiments

In this section, the proposed mathematical model and the solution algorithms are implemented. The mathematical model and the Lagrangian decomposition approach were coded in GAMS optimization software (version 24.1.3) and solved using the CPLEX solver. The hyper-matheuristic algorithms was implemented in MATLAB R2015a and linked with GAMS. All computations were performed on a computer equipped with an Intel(R) Core(TM) i7-3720QM CPU @ 2.6 GHz and 16 GB of RAM.

6.1. A case study

We examine a rice supply chain in Iran as a case study. The supply chain consists of 11 potential suppliers, two rice milling factories as first-stage production centers, and a sorting factory as the second-stage production center. The company also has a warehouse. There are four and three potential locations for establishing new first and second-stage production centers as needed. Additionally, there are four potential locations for establishing warehouses to serve as distribution centers, and the rice is ultimately delivered to nine customer regions. The supply chain process begins at the farms, where it is cultivated and harvested. After harvesting, the rice is sent to the milling factories. Here, the husk is removed, and the rice undergoes initial processing to produce white rice. The processed rice is then transported to the sorting facility. At this stage, the grains are graded based on quality, sorted to remove impurities, and finally packaged for distribution. Once packaged, the final product is sent to the distribution centers, from where it is distributed to the nine customer regions (direct distribution from sorting facilities to customers is also allowed). Fig. 3 illustrates the location of existing facilities and customers on the map. The suppliers and production centers are located in Mazandaran province. The set of potential locations (cities) for establishing new milling and sorting factories and warehouses is presented in Table 6. The ranges of parameters are given in Table 7 based on the data of the case study. The cost parameters are expressed in million tomans, and the parameters

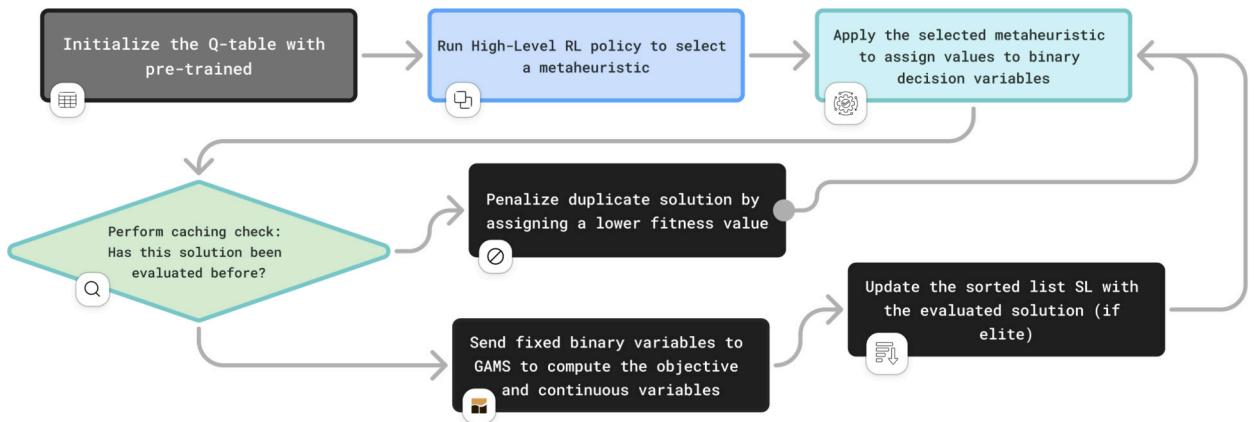


Fig. 2. Procedure of proposed hyper-matheuristic.

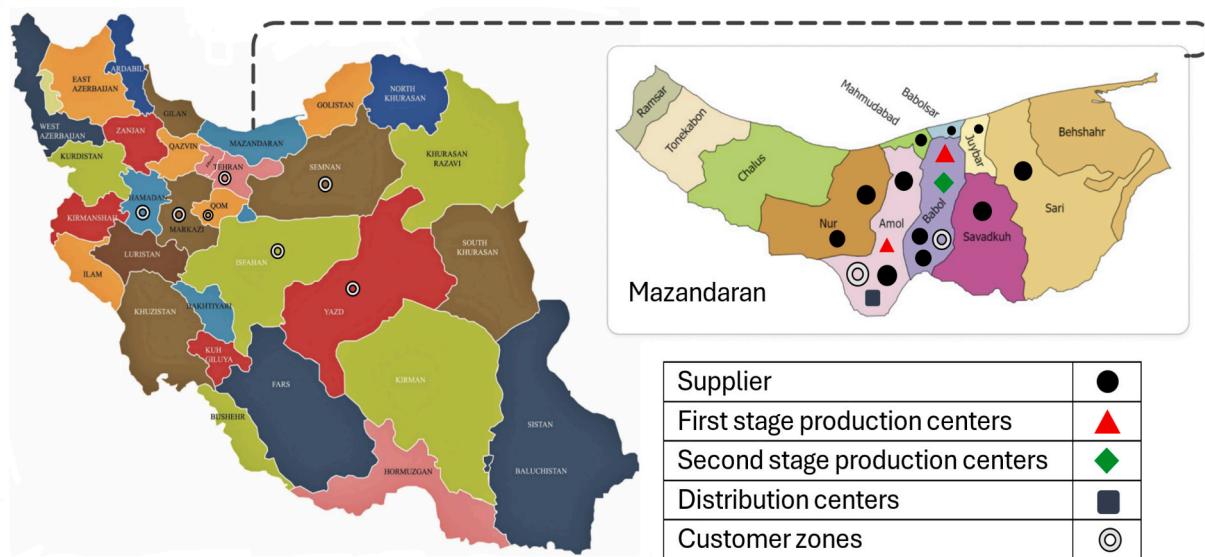


Fig. 3. Locations of facilities and customers.

Table 6
Potential locations for new facilities.

Facility	Potential locations (cities)
Distribution centers	Tehran, Karaj, Esfahan, Qom
First stage production centers	Amol, Babol, Juybar, Mahmudabad
Second stage production centers	Babol, Amol, Babolsar

related to capacities are expressed in tons.

6.2. Evaluation of proposed solution methods

Based on the data of the presented case study, 12 test problems were generated with different numbers of facilities at each level of the supply chain and also different numbers of scenarios. The details of these problems are reported in Table 8.

Before comparing solution methods, the parameter tuning for each metaheuristic algorithm was conducted using the Taguchi design of experiments method, implemented in Minitab 22. The signal-to-noise ratio (S/N) was used as the performance criterion (for more details, see Roy (2010)), with the “smaller-the-better” formulation applied, since the objective was to minimize the performance (objective value) gap between each algorithm and the best-performing method. For SO and AO, the parameter levels were selected based on the values reported in their original publications and refined through trial and error. For GA and PSO, the test ranges were determined based on commonly reported values in the literature, such as those presented by Vali-Siar et al. (2022). Table 9 summarizes the initial values, the selected (tuned) values of the algorithm parameters, and the Taguchi design employed for their calibration.

Table 10 gives the results of running test instances using solution methods. The mathematical model was solved using the CPLEX solver. For a comprehensive performance evaluation of the proposed hyper-matheuristic algorithm, the test problems were also solved using GA as a reference method. The HMH and GA were executed 30 times for each problem. The reason for choosing this method was

Table 7
Values of the parameters of the MILP model.

Parameter	Range/value	Parameter	Range/value
f_{si}	[20, 50]	cap_f	[100, 3000]
fe_k	[1000, 4000]	cad_p	[1000, 2000]
fe_p	[5000, 15000]	Mo_r	[300, 700]
$fe_{p'}$	[3000, 9000]	d_{rc}	[200, 500];
mc_p	[1, 2]	ce_j	[200, 400]
$mc_{p'}$	[0.5, 0.8]	pr_r	[130, 200]
ec_j	[0.5, 2]	δ	[0.75, 0.90]
cr_r	[95, 150]	γ	[0.90, 0.95]
co_r	[110, 180]	λ_{fs}	[0, 1]

Table 8
Generated numerical instances.

No.	Instance size ($ I , R , P , P' , K , C , S $)
1	(3 , 2 , 2 , 2 , 2 , 2 , 4)
2	(6 , 3 , 4 , 4 , 4 , 4 , 6)
3	(8 , 4 , 6 , 6 , 6 , 6 , 8)
4	(12 , 5 , 8 , 8 , 8 , 10 , 10)
5	(24 , 8 , 16 , 16 , 17 , 24 , 22)
6	(26 , 9 , 18 , 18 , 19 , 26 , 23)
7	(28 , 10 , 20 , 20 , 21 , 28 , 24)
8	(30 , 11 , 22 , 22 , 23 , 30 , 25)
9	(50 , 15 , 40 , 40 , 39 , 45 , 40)
10	(53 , 17 , 44 , 44 , 42 , 50 , 43)
11	(57 , 19 , 48 , 48 , 46 , 55 , 47)
12	(60 , 20 , 50 , 50 , 50 , 60 , 50)

that based on testing and solving various test problems, this method outperformed the other methods used in the HMH algorithm. A time limit of 36,000 s was imposed on all solution methods. The columns titled “ t (s)” report the execution time of the methods in seconds. The column titled “Cache” shows the mean percentage of using internal stored data to compute fitness function. For HMH, the best, worst and mean values of the objective function are reported, while for GA the best value is presented. For HMH and GA, the relative percentage deviation is computed, which represents the relative difference between the best found objective value and the best objective value obtained by the solution method ($RPD = (|Method_{sol} - Best_{sol}| / Best_{sol}) \times 100$). LD gives upper bound for problems demonstrating that HMH obtains feasible and near optimal solutions in large-sized problems.

As observed, GAMS is unable to find a solution for instances beyond problem 8 within 36,000 s. Based on the results, HMH outperforms GA and LD in all instances and it has found the optimal solution in four problems. As expected, due to more computations, HMH consumes more time compared to GA, but it gives high-quality solutions which is very valuable. Also, the proposed LD method has been able to obtain high-quality upper bounds.

Fig. 4 depicts the convergence of HMH, GA, and LD for problem 2. As shown by the curve, HMH has obtained high-quality solution in far fewer iterations in comparison with LD. Note that LD has obtained a good upper bound. GA reached its final solution in fewer iterations than LD, which is not optimal.

Fig. 5 shows a box-and-whisker plot of the selection probabilities for four metaheuristics (GA, PSO, SO, AO) over 60 episodes, driven by a pre-trained Q-table. Each box spans the first (Q_1) to third (Q_3) quartiles, with the bold line at the median (Q_2). The interquartile range ($IQR = Q_3 - Q_1$) measures the middle 50 % spread of values. GA has the highest median (~ 0.345) and a wide interquartile range (IQR), indicating that it was the most frequently selected algorithm overall. Although its selection was consistent, there was still some moderate variation across episodes. PSO has a moderate median (~ 0.23) with a large gap between the median and the upper quartile ($Q_2 - Q_3$). This shows that the chance of selecting PSO sometimes increased sharply, meaning the policy explored PSO more during certain episodes. SO has the lowest median and the smallest IQR, along with a consistent downward trend in selection. This suggests that the learning policy gradually reduced its use and eventually ceased relying on this operator. AO has a similar median to PSO (~ 0.26), but the upper part of its selection values is more spread out. This means the chance of selecting AO changed a lot between episodes. Overall, this analysis shows that the high-level reinforcement learning policy balanced exploitation—favoring the consistently strong performer (GA) with exploration, by sometimes using PSO and AO, while gradually phasing out the

Table 9

Initial and tuned parameter values for each algorithm and the Taguchi design used for their calibration.

Algorithm	Parameters	Initial values	Selected value (tuned)	Taguchi design
GA	p_c (crossover rate) p_m (mutation rate) n_{pop} (population size)	0.6, 0.7, 0.8 0.05, 0.10, 0.15 50, 100, 150	0.7 0.10 150	L^9
PSO	c_1 (personal learning coefficient) c_2 (global learning coefficient) ω (inertia weight damping ratio) n_{pop} (population size)	1.0, 1.5, 2.0 1.0, 1.5, 2.0 0.91, 0.95, 0.99 50, 100, 150	1.0 1.5 0.95 100	L^{27}
SO	c_1 (food quantity constant) c_2 (exploration constant) c_3 (exploitation constant) n_{pop} (population size)	0.4, 0.5, 0.6 0.03, 0.05, 0.07 1.0, 2.0, 3.0 50, 100, 150	0.5 0.07 2 100	L^9
AO	α (exploitation adjustment parameter) δ (exploitation adjustment parameter) Nn_{pop} (population size)	0.1, 0.3, 0.5 0.1, 0.3, 0.5 50, 100, 150	0.1 0.3 50	L^9

Table 10

Assessing solution methods.

Instance no.	GAMS		LD		HMH					GA			
	Obj.	t (s)	Obj.	t (s)	Best	Worst	Mean	t (s)	RPD	Cache	Best	t (s)	RPD
1	2027.56	32.01	2027.56	24.34	2027.56	1929.23	2001.27	30.71	0.00 %	0.82	2027.56	8.15	0.00 %
2	3221.29	41.26	3226.49	40.71	3221.49	3081.55	3191.08	35.89	0.00 %	0.78	3001.82	11.71	6.82 %
3	9363.84	53.17	9851.89	73.18	9363.84	8895.34	9026.11	41.65	0.00 %	0.76	8714.51	15.36	6.93 %
4	12049.03	71.20	12620.73	96.14	12049.03	11704.91	11953.12	60.12	0.00 %	0.72	10996.12	21.76	8.74 %
5	9276.10	426.12	11192.67	1527.53	9211.14	8114.27	8632.98	348.87	0.70 %	0.63	8014.74	210.10	12.99 %
6	39402.08	925.54	42514.51	2156.90	38547.18	35974.10	37162.12	429.04	2.17 %	0.65	33610.18	271.58	14.70 %
7	33341.64	1832.94	37851.15	3821.41	31984.72	28905.38	30541.10	490.31	4.07 %	0.62	27162.19	321.14	18.53 %
8	31844.47	1432.91	36756.97	4022.17	30610.37	26879.70	28640.91	537.22	3.88 %	0.64	25360.83	396.31	20.36 %
9	—	>36000	63471.12	>36000	56218.91	51271.06	54708.66	2948.87	0.00 %	0.56	47891.50	1371.85	14.81 %
10	—	>36000	67812.58	>36000	61277.03	56691.74	58095.48	3219.74	0.00 %	0.52	51840.67	1497.18	15.40 %
11	—	>36000	62176.95	>36000	55491.12	50071.83	52796.32	3791.06	0.00 %	0.53	48025.79	1799.53	13.45 %
12	—	>36000	69280.59	>36000	60173.46	53890.33	56901.43	4380.47	0.00 %	0.48	49991.70	2015.02	16.92 %
Average	—	—	37886.88	—	30847.99	28117.45	29470.88	1359.49	0.90 %	0.64	26385.47	661.64	12.47 %

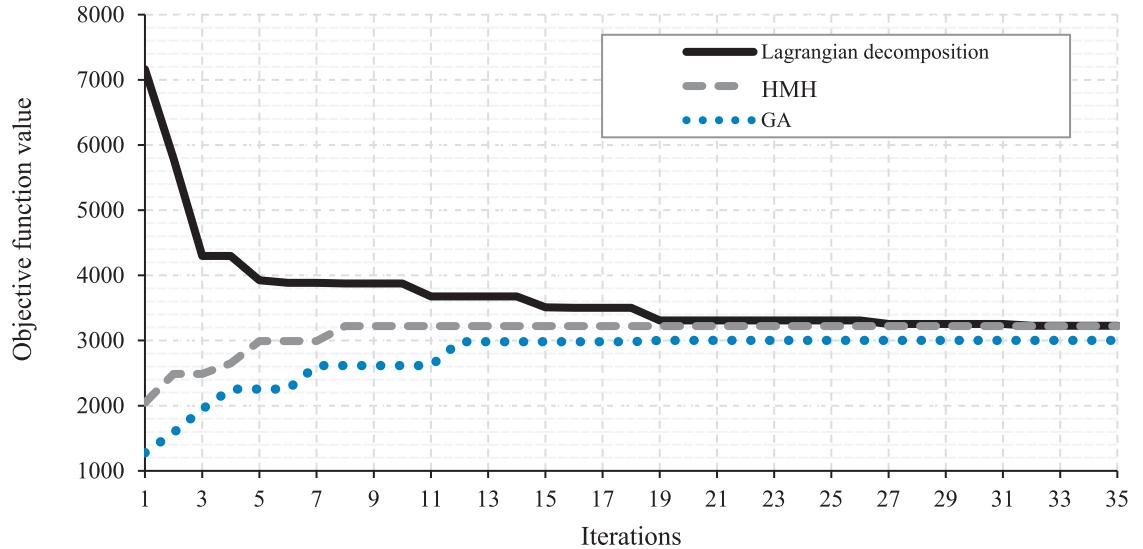


Fig. 4. Comparison of the convergence of solution methods.

underperforming operator (SO).

6.3. Implementing theories and analysis

Consider that the described SC of the case study is under random disruptions. Fig. 6 portrays a set of 60 data points representing the inter-arrival times between disruptive events, or shocks, and the associated damages for one of the first stage production centers. As illustrated, an analysis of disruption data shows a clear dependency between the inter-arrival time of shocks and the damage severity at facilities. Specifically, when shocks occur in rapid succession (short inter-arrival times), the system has insufficient recovery time, which leads to more severe capacity losses. We adopted GA-based clustering method, as proposed by Shamstabar et al. (2024) for handling the dependency between inter-arrival times of shocks and their damage magnitudes in reliability modeling. Classical clustering methods were unsuitable for this study because their typical spherical or convex cluster geometries do not align with the structure of the data (Han et al. 2022). To implement GA-based clustering, the maximum number of iterations as the stopping criterion was selected to be 100. Different values were tested to determine the number of clusters. According to Fig. 7, the value of the sum of the squared distance of the data points from their closest cluster center decreases with increasing number of clusters (r). The elbow method can be applied to specify the optimal number of clusters (Mouton et al., 2020). The largest difference identifies the elbow point. The total within-clusters sum of squares was obtained using the proposed metaheuristic clustering algorithm. The outcomes indicate that the optimal number of clusters is three.

Based on the selected number of clusters and using the metaheuristic algorithm, the dataset was clustered. The outcome is shown in Fig. 8.

As described above, the *PH* distributions of the clustered data are needed. To this end, the expectation–maximization (EM) algorithm for fitting *PH* distributions proposed by Thummler et al. (2006) was used. The data on damage caused by disruptive events in cluster 1 (W_{1j}) has $PH(\omega_{1j}, \mathbf{W}_{1j})$ with the following parameters:

$$\omega_{1j} = [6.0562e - 07, 0.9999, 0, 0, 0, 0, 0, 0, 0, 0]_{1 \times 10},$$

$$\mathbf{W}_{1j} = \begin{bmatrix} -8.9890e - 05 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & -7.4937e - 04 & 7.4937e - 04 & 0 & 0 & \dots & 0 \\ 0 & 0 & -7.4937e - 04 & 7.4937e - 04 & 0 & \dots & 0 \\ 0 & 0 & 0 & -7.4937e - 04 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & -7.4937e - 04 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & -7.4937e - 04 \end{bmatrix}_{10 \times 10}.$$

The damage caused by disruptive events data in cluster 2 (W_{2j}) has $PH(\omega_{2j}, \mathbf{W}_{2j})$ with following parameters:

$$\omega_{2j} = [2.9442e - 07, 0.9999, 0, 0, 0, 0, 0, 0, 0, 0]_{1 \times 12},$$

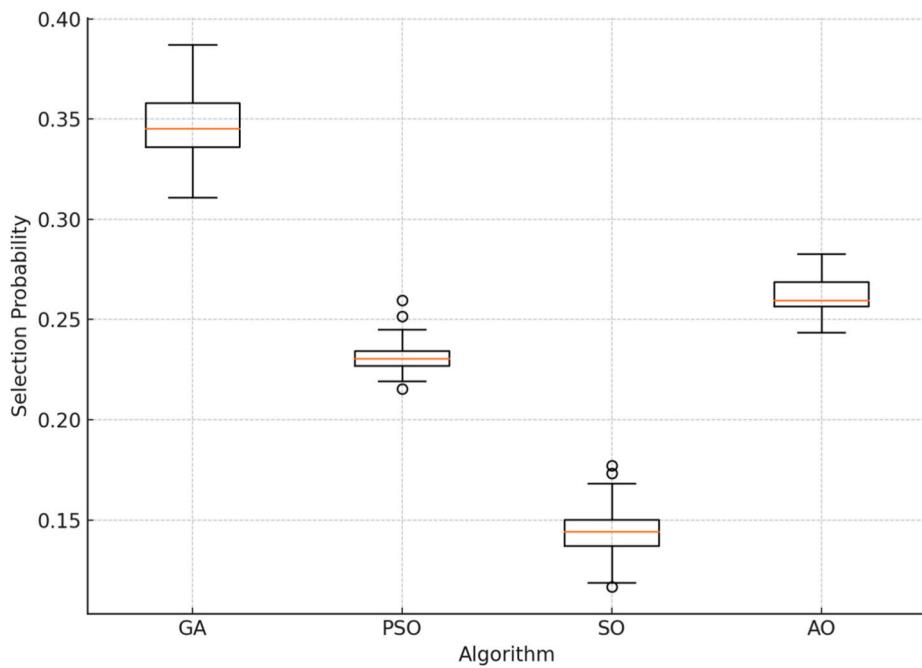


Fig. 5. Boxplot of selection probabilities per algorithm.

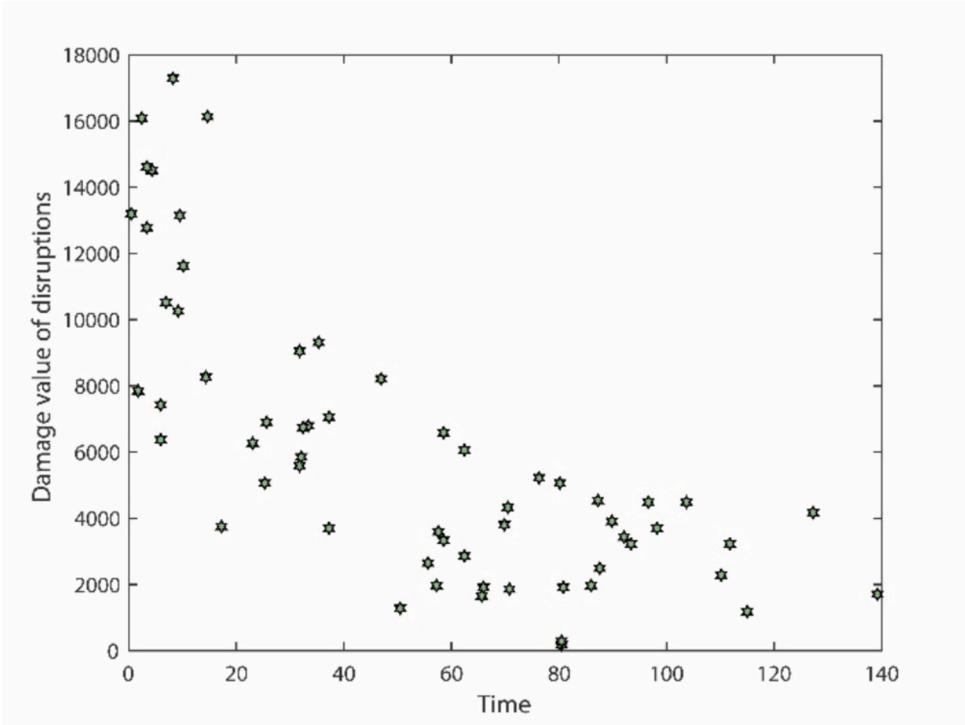


Fig. 6. The inter-arrival times between disruptive events and their damages.

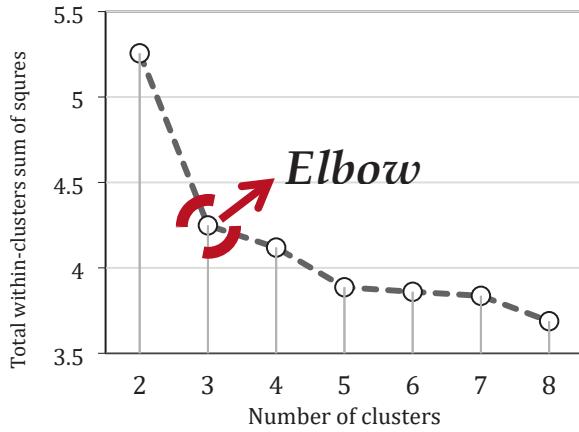


Fig. 7. Total within-clusters sum of squares for different number of clusters.

$$\mathbf{W}_{2j} = \begin{bmatrix} -1.6658e - 04 & 0 & 0 & 0 & \dots & 0 \\ 0 & -0.0017 & 0.0017 & 0 & \dots & 0 \\ 0 & 0 & -0.0017 & 0.0017 & \ddots & \vdots \\ 0 & 0 & 0 & \ddots & \ddots & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & 0.0017 \\ 0 & 0 & 0 & 0 & 0 & -0.0017 \end{bmatrix}_{12 \times 12}.$$

And the damage caused by disruptive events data in cluster 3 (\mathbf{W}_{3j}) has $PH(\omega_{3j}, \mathbf{W}_{3j})$ with following parameters:

$$\omega_{3j} = [0.0621, 0, 0, 0.9379, 0, 0, 0, 0]_{1 \times 8},$$

$$\mathbf{W}_{3j} = \begin{bmatrix} -0.0134 & 0.0134 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.0134 & 0.0134 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.0134 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.0015 & 0.0015 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.0015 & 0.0015 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.0015 & 0.0015 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.0015 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0015 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.0015 \end{bmatrix}_{8 \times 8}$$

Using the theories presented in Section 4, the approximate damage distribution was obtained. These steps were then applied to obtain the distribution of damages for other facilities. In the next step, Monte Carlo simulation was applied to generate disruption scenarios. Subsequently, the proposed scenario reduction method was applied.

To evaluate the robustness of the proposed scenario generation and reduction procedure, we applied the in-sample stability test (Kaut and Wallace, 2007). This test measures the variability of the optimal objective value when the model is solved repeatedly on different scenario sets generated from the same distribution. In detail, the robustness of the proposed procedure was evaluated by varying the number of initial scenarios O and the number of retained scenarios (after reduction) S . For each $O \in \{100, 200, 300, 400\}$ and $S \in \{6, 8, 10, 12\}$, 10 independent scenario sets were generated, reduced using the FCM, and applied in the two-stage stochastic program. Table 11 presents the mean (μ) and standard deviation (SD (σ)) of the optimal expected profit across replications, along with the relative SD (σ/μ). The results show that as O increases, the SD decreases, indicating more stable solutions across random draws. Moreover, in all cases the relative SD remains well below 2 % (see for example Backe et al., 2021), demonstrating that the proposed method achieves a high degree of in-sample stability and robustness.

Based on the computational results, a retained scenario set size of 10 was selected for the case study problem, as it provided a satisfactory balance between solution stability (relative SD well below 1 %) and computational efficiency. Also, increasing the number of retained scenarios substantially raises the computational complexity and solution time, while the stability gains beyond this point are marginal. Furthermore, the relative SD values suggest that the number of generated scenarios should preferably not be less than 300 to ensure robust results. The same computations were done for problem instances of the previous subsection. The fuzzifier parameter (the weighting exponent m) in FCM was set to $m = 2$, which is the most commonly used value in the literature (Pal and Bezdek, 1995). Based on our computations, varying m around this value did not lead to any significant changes in the results.

The described company has decided to fortify its SC against disruptions. As mentioned earlier, the company has considered some potential locations to increase production capacity, meet additional demand, and achieve higher profits by developing its SC facilities. The presented mathematical model was solved under two conditions: resilient and non-resilient modes. In the latter, no resilience strategies are implemented. The results are reported in Table 12. According to the results, the profit of the SC in resilient mode is

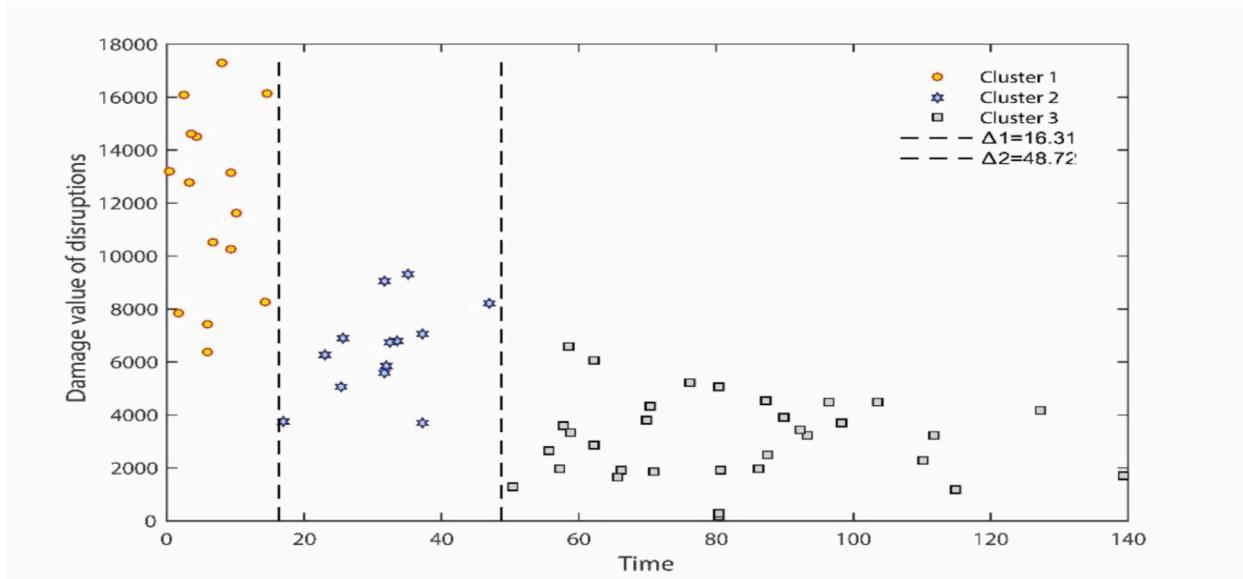


Fig. 8. Data clustering.

Table 11

The results of investigating the robustness of the proposed scenario generation and reduction procedure.

Number of generated Scenarios (N)	Number of retained scenarios (S)	Mean (μ)	SD (σ)	Relative SD (σ/μ)
100	6	25077.36	497.56	1.98 %
	8	23833.40	380.96	1.60 %
	10	25165.73	298.71	1.19 %
	12	25286.92	288.05	1.14 %
200	6	25238.01	445.16	1.76 %
	8	25264.07	339.17	1.34 %
	10	25296.20	260.55	1.03 %
	12	25454.06	254.90	1.00 %
300	6	25203.26	337.12	1.34 %
	8	25190.48	214.43	0.85 %
	10	25318.19	150.19	0.59 %
	12	25087.08	143.12	0.57 %
400	6	25298.45	308.54	1.22 %
	8	25695.89	197.72	0.77 %
	10	25358.22	137.63	0.54 %
	12	25448.41	134.38	0.53 %

approximately 182 % more than the profit of the non-resilient SC. The non-resilient SC has missed more than 60 % of the potential demand. The SC in resilient mode has been substantially developed. Four production centers and three DCs have been established, whereas in non-resilient SC, development is not impressive.

This schematic diagram in Fig. 9 illustrates the flow of products between the nodes of the supply chain for scenario 1 as an example. The thicker links indicate higher volume of flows.

To analyze the impact of resilience strategies, three levels of disruption —low (0–33 % decrease in capacity of facilities), medium (34–66 % decrease in capacity of facilities), and severe (67–100 % decrease in capacity of facilities)—are considered. It is evident that as the severity of the disruption increases, facility capacity declines further. As previously mentioned, this study utilizes resilience strategies including multiple sourcing (MS), dual-channel distribution (DD), capacity expansion (CE) and outsourcing (OU). In Fig. 10, each time, one of the strategies (whose abbreviation is provided below the corresponding chart) is implemented, and the objective function value is presented. Additionally, the figures show the objective function value for the non-resilient (NR) case, where no resilience strategy is applied.

When the disruption is at a low level, the multiple sourcing strategy has improved the objective function by up to 93 % compared to non-resilient mode. The outsourcing strategy ranks next with approximately 41 %, followed by dual-channel distribution at 16 % and

capacity expansion at 15 %. At the medium disruption level, the ranking of strategies remains the same as before, but their impact has increased. The multiple sourcing strategy has improved profitability by 160 %, outsourcing by 81 %, dual-channel distribution by 28 %, and capacity expansion by 26 % compared to the non-resilient mode. When the disruption is at a high level, the supply chain's profit in the non-resilient case is nearly zero, and the resilience strategies have effectively mitigated the disruptions. Among them, the most significant impact comes from the outsourcing strategy, which has increased the supply chain's profit by approximately 55-fold. The effects of other strategies are as follows: multiple sourcing: 390 %, dual-channel distribution: 759 %, and capacity expansion: 6 %.

The presented diagrams implicitly provided a sensitivity analysis on the severity of disruptions. When the severity of disruptions or equivalently their damaging effects increase, the objective will deteriorate and also the impact of strategies is highlighted further. Fig. 11 illustrates the synergy effect of resilience strategies under medium-level disruptions. Based on the results, the simultaneous use of resilience strategies leads to significantly greater improvements compared to using a single strategy alone. When sourcing is limited to a single supplier, and a disruption prevents a substantial upstream flow in the supply chain, strategies like capacity expansion or dual-channel distribution cannot generate a significant improvement. This is because these strategies are only effective when material flow from the supplier reaches production centers. However, if multiple sourcing is implemented, the aforementioned strategies can then be effective. This chart can also be depicted and analyzed for cases where two strategies are used together. The highest improvement is observed when all strategies are applied simultaneously, resulting in a 214 % increase in profit.

In the final part of this section, some sensitivity analyses are conducted on some key parameters of the problem. Fig. 12 presents the sensitivity analysis on two key parameters: demand and capacity. The first sensitivity analysis examines the impact of demand fluctuations (expressed as a percentage) on the objective function. As observed, a decrease in demand leads to a decline in the supply chain's profit, and as expected, when demand drops to zero, the objective function also becomes zero. Conversely, as demand increases, profit rises. In the resilient supply chain, profit continues to grow up to an 80 % increase in demand, after which it remains constant. This could be due to two reasons: a) Full utilization of potential capacity, meaning the system can no longer meet additional demand. b) The establishment of new facilities is not economically justified. In the non-resilient mode, the SC profit is less and after a 60 % increase, profit remains constant.

The second sensitivity analysis investigates the impact of the capacity of the first-stage production centers. As expected, a decrease in capacity leads to a decrease in profit, while an increase in the capacity results in an upward trend in profit. However, the rate of profit increase diminishes. In the non-resilient supply chain, when capacity is set to zero, the profit drops to zero, indicating its fragility. Conversely, in the resilient supply chain, the outsourcing strategy prevents the profit from becoming zero, since purchasing rice from competitors is cost-effective.

Fig. 13 illustrates the impact of variations in key cost parameters on the objective value. As expected, increasing costs lead to a deterioration in the objective function value. For each cost parameter, the value was increased by 25 % (where Case I represents the baseline scenario (no increase)). The results indicate that the unit manufacturing cost has the greatest influence on the objective value, followed by establishment costs and shipment costs as the second and third most influential parameters, respectively.

To evaluate the added value of the proposed DDSSP, we conducted a comparison against a simpler baseline approach, namely the traditional scenario-based stochastic programming (SSP). In the baseline method, disruption scenarios were generated using three common parametric distributions (exponential, normal, and uniform), without considering inter-arrival time dependency or using a data-driven approach. Eight realizations were generated based on the data of the case study and treated as representing true future conditions. The first column in Table 13 shows the deterministic solution where all disruption data are known in advance, providing an upper bound for the objective function value. For each of the SSP and DDSSP methods, we fixed the value of the first-stage variables obtained from that method in the deterministic model, and evaluated the resulting objective value.

As shown, because the traditional SSP does not capture inter-arrival time dependencies or the structure of the real disruption data, it produced infeasible solutions in two out of eight realizations and achieved lower profits in the remaining instances. For a fair comparison, scenarios in the SSP method were generated separately using each of the three mentioned distributions, and for each realization, the best solution among these was reported. The DDSSP consistently provided feasible solutions and outperformed SSP, demonstrating the value of incorporating data-driven scenario-based modeling. The last column of the table shows the improvement achieved by DDSSP, calculated as the difference between the objective values obtained by DDSSP and SSP.

6.4. Managerial insights

The proposed resilient SCND problem was solved and analyzed based on a case study from the rice SC. However, the models and solution methods presented in this study can be applied to other industries, such as pharmaceuticals, food, and various manufacturing sectors. The findings indicate that supply chain resilience plays a crucial role in achieving SC objectives. The presented results demonstrated a considerable impact of resilience strategies on SC profit. Managers can mitigate the impact of disruptive events on the SC network by employing one or more of the proposed resilience strategies. Results showed that resilience strategies can improve profit by about 214 % in comparison with the non-resilient mode.

An important aspect preceding the selection and implementation of resilience strategies is the anticipation, identification, and estimation of the severity of their impacts. Disruptions originate from different sources like natural disasters, pandemics, human errors, shocks, and failures of facilities. However, determining the impacts of disruptions is a big challenge. To cope with this issue, industry experts can utilize the presented *PH*-disruption model and available data to assess the probability distribution of damages resulting

Table 12

The results of solving the case study problem under resilient and non-resilient conditions.

	Resilient supply chain	Non-resilient supply chain
Objective function value (profit)	25301.94	8959.18
Selected suppliers	Amol, Babol, Babolsar	Amol
New established first stage production center	Amol	—
New established second stage production centers	Babol, Amol, Babolsar	Babol
New established distribution centers	Tehran, Karaj	Tehran
Percentage of expected satisfied demand	97 %	38 %

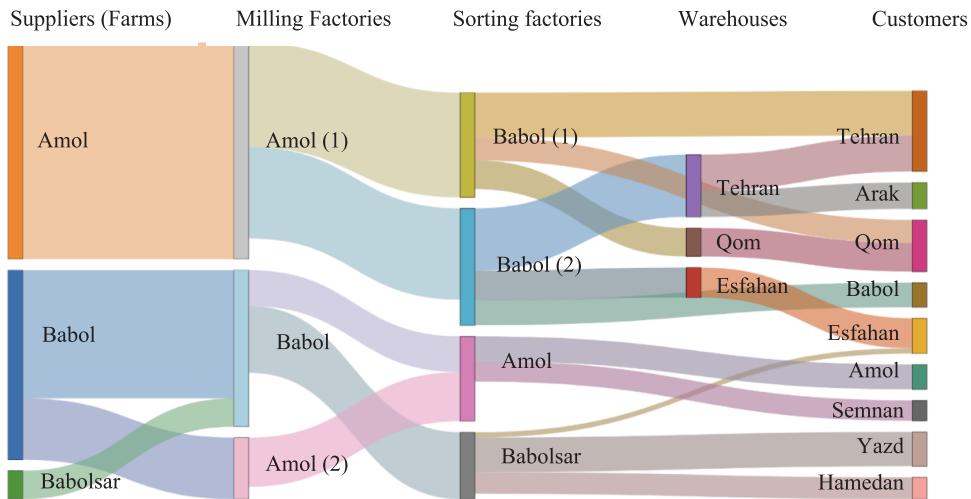


Fig. 9. The flows between supply chain nodes (scenario 1).

from disruptive events and generate disruption scenarios more accurately. Subsequently, employing a scenario-based stochastic programming method, they can engage in the proper design of the SC network. Especially in situations where super-disruptions occur and ripple effects manifest within the SC, the utilization of relevant methods is highly crucial for proper planning, since the frequent occurrence of disruptive events can threaten the survival of the company and the SC. Engineers should pay attention to the uncertainties of the parameters involved in the SCND problem and use proper methods such as stochastic programming to deal with these inherent risks. The results demonstrated the importance of some critical parameters such as capacity, manufacturing cost and demand. These parameters must be tuned and determined carefully before starting to optimize the network. Decreasing the costs of production, accurately forecasting and identifying demand and developing the suitable capacity are crucial, as the objective function is highly sensitive to these parameters.

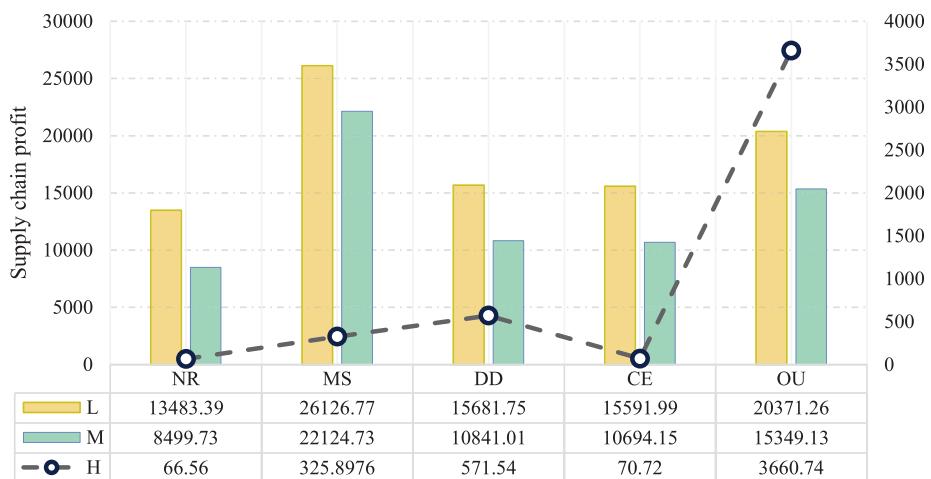


Fig. 10. Investigating the effect of resilience strategies on supply chain objective under different disruption levels.

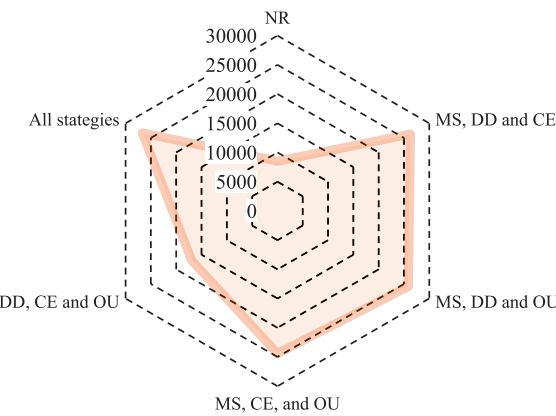


Fig. 11. Examining the simultaneous use of resilience strategies (synergy effect).

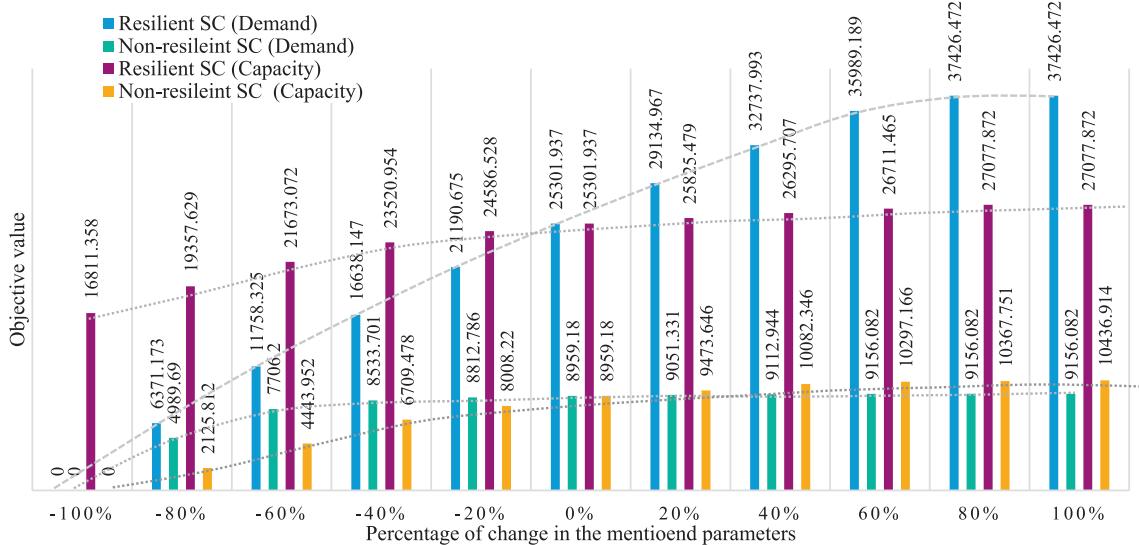


Fig. 12. Examining the effect of changing demand and capacity on the objective value.

7. Conclusions

Disruptions have been shown to bring a profound influence on all dimensions of SCs, particularly their economic performance. This study investigated the resilient SCND problem under super-disruptions. Notably, we explore the relationship between the inter-arrival time of disruptive events and their impacts on the SC design. Moreover, we address the ripple effect through this innovative approach.

To tackle these challenges, we propose a two-phase hybrid methodology aimed at designing a resilient SC. The initial phase of the developed approach incorporated a clustering method capable of identifying patterns of dependency within the dataset with regard to disruptive events. First, a GA clustering algorithm was proposed to implement the clustering. Then the *PH* distributions and related theorems were used to obtain the probability distributions of disruption based on the clusters. After that, the Monte Carlo simulation method was applied for generating disruption scenarios using the found distributions. To decrease computational complexity, the fuzzy c-means clustering was used to decrease the number of scenarios. We developed a new two-stage stochastic MILP model to design the SC network and applied Lagrangian decomposition, a hyper-matheuristic using reinforcement learning and the genetic algorithm to solve the model. Four resilience strategies were used to improve SC resilience. A case study was presented in the rice industry to show the effectiveness of the proposed methods. The results demonstrated that the proposed methodology, including machine learning methods, the *PH*-disruption model, and the optimization model, is effective and can be applied to design a resilient SC. Furthermore, resilience strategies have been shown to be very efficient and can significantly improve objective function especially when they are used simultaneously.

Considering uncertainty in other parameters of the problem such as demand and costs and applying suitable optimization methods based on the type of uncertainty represents a valuable avenue for future research. This could include developing a data-driven robust optimization model, which itself would be a novel contribution. Another promising direction is to extend the current single-period

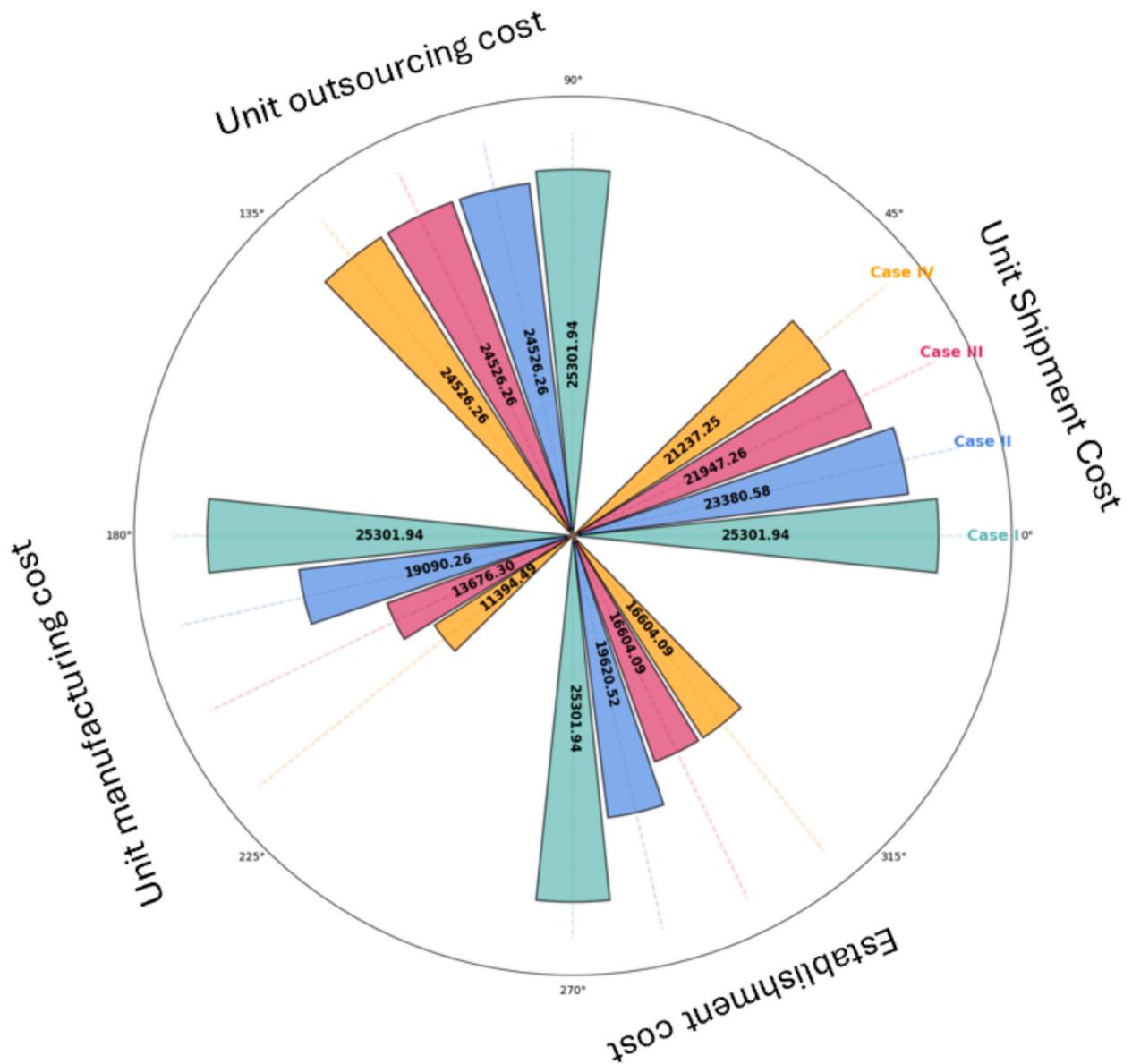


Fig. 13. Analyzing the impact of changing costs on the objective value.

Table 13

Comparing the proposed DDSSP and the traditional SSP based on the objective function.

Realization No.	Deterministic (no uncertainty)	SSP	DDSSP	Value of DDSSP solution
1	25612.08	19845.52	24587.91	4742.39
2	25433.73	20328.26	24781.83	4453.57
3	25689.16	Infeasible	25134.38	--
4	24963.98	20122.33	24216.47	4094.14
5	25547.51	20284.14	24962.06	4677.92
6	24906.28	Infeasible	24331.73	--
7	25412.39	20564.58	24794.62	4230.04
8	25073.04	19978.26	24472.76	4494.50
Average	25329.77	20187.18	24660.22	4448.76

formulation to a multi-period setting and incorporate multi-stage stochastic programming to more effectively capture uncertainty over time. Further investigations could also explore additional tactical and operational decisions, such as pricing and routing, by integrating them into the mathematical model. Finally, developing more efficient exact solution methods for solving the proposed model remains an important line of research.

CRediT authorship contribution statement

Mohammad Mahdi Vali-Siar: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Conceptualization. **Hamid Tikani:** Writing – review & editing, Visualization, Investigation, Conceptualization. **Emrah Demir:** Writing – review & editing, Supervision, Investigation, Conceptualization. **Yousof Shamstabar:** Writing – review & editing, Validation, Methodology, Investigation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We would like to extend our sincere gratitude to the Editors and Reviewers for their invaluable contributions to this manuscript. Their expertise, insights, and constructive feedback have significantly enhanced the quality and presentation of our work. This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Data availability

The authors do not have permission to share data.

References

- Abualigah, L., Yousri, D., Abd Elaziz, M., Ewees, A.A., Al-Qaness, M.A., Gandomi, A.H., 2021. Aquila optimizer: a novel meta-heuristic optimization algorithm. *Comput. Ind. Eng.* 157, 107250.
- Alhasawi, E., Hajli, N. and Dennehy, D., 2023, September. A Review of Artificial Intelligence (AI) and Machine Learning (ML) for Supply Chain Resilience: Preliminary Findings. In *2023 IEEE International Symposium on Technology and Society (ISTAS)* (pp. 1-8). IEEE.
- Alikhani, R., Ranjbar, A., Jamali, A., Torabi, S.A., Zobel, C.W., 2023. Towards increasing synergistic effects of resilience strategies in supply chain network design. *Omega* 116, 102819.
- Azad, N., Saharidis, G.K., Davoudpour, H., Malekly, H., Yektamaram, S.A., 2013. Strategies for protecting supply chain networks against facility and transportation disruptions: an improved Benders decomposition approach. *Ann. Oper. Res.* 210 (1), 125–163.
- Backe, S., Ahang, M., Tomassgard, A., 2021. Stable stochastic capacity expansion with variable renewables: comparing moment matching and stratified scenario generation sampling. *Appl. Energy* 302, 117538.
- Bagirov, A.M., Karmitsa, N., Taheri, S., 2020. Partitional Clustering via Nonsmooth Optimization: Clustering via Optimization. Springer Nature.
- Belhadi, A., Kamble, S., Fosso Wamba, S., Queiroz, M.M., 2022. Building supply-chain resilience: an artificial intelligence-based technique and decision-making framework. *Int. J. Prod. Res.* 60 (14), 4487–4507.
- Bezdek, J.C., Ehrlich, R., Full, W., 1984. FCM: The fuzzy c-means clustering algorithm. *Computers & Geosci.* 10 (2–3), 191–203.
- Bladt, M., Nielsen, B.F., 2017. Matrix-Exponential Distributions in Applied Probability, Vol. 81. Springer, New York.
- Camur, M.C., Ravi, S.K., Saleh, S., 2024. Enhancing supply chain resilience: a machine learning approach for predicting product availability dates under disruption. *Expert Syst. Appl.* 247, 123226.
- Christopher, M., Peck, H., 2004. Building the resilient supply chain. *Int. J. Logist. Manaeg.* 15 (2), 1–14.
- Esmaeli, F., Mafakheri, F., Nasiri, F., 2023. Biomass supply chain resilience: Integrating demand and availability predictions into routing decisions using machine learning. *Smart Sci.* 11 (2), 293–317.
- Fattah, M., Govindan, K., Keyvanshokooh, E., 2017. Responsive and resilient supply chain network design under operational and disruption risks with delivery lead-time sensitive customers. *Transp. Res. Part E: Logist. Transp. Rev.* 101, 176–200.
- Foroozesh, N., Karimi, B., Mousavi, S.M., Mojtabaei, M., 2023. A hybrid decision-making method using robust programming and interval-valued fuzzy sets for sustainable-resilient supply chain network design considering circular economy and technology levels. *J. Ind. Inf. Integr.* 33, 100440.
- Gholami-Zanjani, S.M., Klibi, W., Jabalameli, M.S., Pishvaee, M.S., 2021. The design of resilient food supply chain networks prone to epidemic disruptions. *Int. J. Prod. Econ.* 233, 108001.
- Govindan, K., Paam, P., Abtahi, A.R., 2016. A fuzzy multi-objective optimization model for sustainable reverse logistics network design. *Ecol. Ind.* 67, 753–768.
- Hasani, A., Khosrojerdi, A., 2016. Robust global supply chain network design under disruption and uncertainty considering resilience strategies: a parallel memetic algorithm for a real-life case study. *Transp. Res. Part E: Logist. Transp. Rev.* 87, 20–52.
- Hasani, A., Mokhtari, H., Fattah, M., 2021. A multi-objective optimization approach for green and resilient supply chain network design: a real-life case study. *J. Clean. Prod.* 278, 123199.
- Hashim, F.A., Hussien, A.G., 2022. Snake optimizer: a novel meta-heuristic optimization algorithm. *Knowl.-Based Syst.* 242, 108320.
- Hohenstein, N.O., Feisel, E., Hartmann, E., Giunipero, L., 2015. Research on the phenomenon of supply chain resilience: a systematic review and paths for further investigation. *Int. J. Phys. Distribut. Logist. Manage.* 45 (1/2), 90–117.
- Holland, J.H., 1975. Adaption in Natural and Adaptive Systems. University of Michigan Press, Ann Arbor.
- Hao, X., Demir, E., 2023. Artificial intelligence in supply chain decision-making: an environmental, social, and governance triggering and technological inhibiting protocol. *J. Model. Manage.*
- Hao, X., Demir, E., 2024. Artificial intelligence in supply chain management: enablers and constraints in pre-development, deployment, and post-development stages. *Product. Plann. Control* 1–23.
- Han, J., Pei, J., Tong, H. (Eds.), 2022. Data Mining: Concepts and Techniques. Morgan Kaufmann.
- Jafarian, M., Mahdavi, I., Tajdin, A., Tirkolaei, E.B., 2025. A multi-stage machine learning model to design a sustainable-resilient-digitalized pharmaceutical supply chain. *Socio-Econ. Plann. Sci.*, 102165

- Ivanov, D., 2021. Introduction to Supply Chain Resilience: Management, Modelling, Technology. Springer Nature.
- Jabbarzadeh, A., Fahimnia, B., Sheu, J.B., Moghadam, H.S., 2016. Designing a supply chain resilient to major disruptions and supply/demand interruptions. *Transp. Res. B Methodol.* 94, 121–149.
- Jabbarzadeh, A., Fahimnia, B., Sabouhi, F., 2018. Resilient and sustainable supply chain design: sustainability analysis under disruption risks. *Int. J. Prod. Res.* 56 (17), 5945–5968.
- Kaut, M., Wallace, S.W., 2007. Evaluation of scenario-generation methods for stochastic programming. *Pacific J. Optim.* 3 (2), 257–271.
- Kennedy, J., Eberhart, R., 1995, November. Particle swarm optimization. In: *Proceedings of ICNN'95-international conference on neural networks* (Vol. 4, pp. 1942–1948). IEEE.
- Liu, M., Lin, T., Chu, F., Ding, Y., Zheng, F., Chu, C., 2023. Bi-objective optimization for supply chain ripple effect management under disruption risks with supplier actions. *Int. J. Prod. Econ.* 265, 108997.
- Mak, H.Y., Shen, Z.J., 2012. Risk diversification and risk pooling in supply chain design. *IIE Trans.* 44 (8), 603–621.
- Mouton, J.P., Ferreira, M., Helberg, A.S., 2020. A comparison of clustering algorithms for automatic modulation classification. *Expert Syst. Appl.* 151, 113317.
- Melniky, S.A., Rodrigues, A., Ragatz, G.L., 2009. Using simulation to investigate supply chain disruptions. In: *Supply Chain Risk: A Handbook of Assessment, Management, and Performance*. Springer, US, Boston, MA, pp. 103–122.
- Murphy, K.P., 2012. Machine Learning: A Probabilistic Perspective. MIT Press.
- Nakagawa, T., 2007. Shock and Damage Models in Reliability Theory. Springer Science & Business Media.
- Namdar, J., Li, X., Sawhney, R., Pradhan, N., 2018. Supply chain resilience for single and multiple sourcing in the presence of disruption risks. *Int. J. Prod. Res.* 56 (6), 2339–2360.
- Nili, M., Jabalameli, M.S., Jabbarzadeh, A., Dehghani, E., 2025. An optimization approach for sustainable and resilient closed-loop floating solar photovoltaic supply chain network design. *Comput. Chem. Eng.* 193, 108927.
- Pal, N.R., Bezdek, J.C., 1995. On cluster validity for the fuzzy c-means model. *IEEE Trans. Fuzzy Syst.* 3 (3), 370–379.
- Ranjkesh, S.H., Hamadani, A.Z., Mahmoodi, S., 2019. A new cumulative shock model with damage and inter-arrival time dependency. *Reliab. Eng. Syst. Saf.* 192, 106047.
- Riascos-Ochoa, J., Sanchez-Silva, M., Akhavan-Tabatabaei, R., 2014. Reliability analysis of shock-based deterioration using phase-type distributions. *Probab. Eng. Mech.* 38, 88–101.
- Roozkosh, P., Pooya, A., Agarwal, R., 2023. Blockchain acceptance rate prediction in the resilient supply chain with hybrid system dynamics and machine learning approach. *Oper. Manage. Res.* 16 (2), 705–725.
- Roy, R.K. (Ed.), 2010. A primer on Taguchi method. Society of Manufacturing Engineers, Michigan.
- Sabouhi, F., Pishvaei, M.S., Jabalameli, M.S., 2018. Resilient supply chain design under operational and disruption risks considering quantity discount: a case study of pharmaceutical supply chain. *Comput. Ind. Eng.* 126, 657–672.
- Sabouhi, F., Jabalameli, M.S., Jabbarzadeh, A., Fahimnia, B., 2020. A multi-cut L-shaped method for resilient and responsive supply chain network design. *Int. J. Prod. Res.* 58 (24), 7353–7381.
- Sabouhi, F., Jabalameli, M.S., Jabbarzadeh, A., 2021. An optimization approach for sustainable and resilient supply chain design with regional considerations. *Comput. Ind. Eng.* 159, 107510.
- Salehi Sadghiani, N., Torabi, S.A., Sahebjamnia, N., 2015. Retail supply chain network design under operational and disruption risks. *Transp. Res. Part E: Logist. Transp. Res.* 75, 95–114.
- Sazvar, Z., Tafakkori, K., Oladzad, N., Nayeri, S., 2021. A capacity planning approach for sustainable-resilient supply chain network design under uncertainty: a case study of vaccine supply chain. *Comput. Ind. Eng.* 159, 107406.
- Sawik, T., 2022. Stochastic optimization of supply chain resilience under ripple effect: a COVID-19 pandemic related study. *Omega* 109, 102596.
- Sawik, T., 2023. A stochastic optimisation approach to maintain supply chain viability under the ripple effect. *Int. J. Prod. Res.* 61 (8), 2452–2469.
- Shamstabar, Y., Shahriari, H., Samimi, Y., 2021. Reliability monitoring of systems with cumulative shock-based deterioration process. *Reliab. Eng. Syst. Saf.* 216, 107937.
- Shamstabar, Y., Ebrahimmaghah, N., Samimi, Y. and Shahriari, H., 2024. Reliability modeling of systems with random failure threshold subjected to cumulative shocks using machine learning method. *Quality and Reliability Engineering International*, article in press.
- Sindhwani, R., Jayaram, J., Saddikuti, V., 2023. Ripple effect mitigation capabilities of a hub and spoke distribution network: an empirical analysis of pharmaceutical supply chains in India. *Int. J. Prod. Res.* 61 (8), 2795–2827.
- Tang, C.S., 2006. Perspectives in supply chain risk management. *Int. J. Prod. Econ.* 103 (2), 451–488.
- Thummler, A., Buchholz, P., Telek, M., 2006. A novel approach for phase-type fitting with the EM algorithm. *IEEE Trans. Dependable Secure Comput.* 3 (3), 245–258.
- Tikani, H., Setak, M., 2019. Efficient solution algorithms for a time-critical reliable transportation problem in multigraph networks with FIFO property. *Appl. Soft Comput.* 74, 504–528.
- Tsao, Y.C., Balo, H.T., Lee, C.K.H., 2024. Resilient and sustainable semiconductor supply chain network design under trade credit and uncertainty of supply and demand. *Int. J. Prod. Econ.* 274, 109318.
- Vali-Siar, M.M., Roghanian, E., 2020. Resilient mixed supply chain network redesign under operational and disruption risks: a case study. *J. Ind. Eng. Res. Prod. Syst.* 8 (16), 113–135.
- Vali-Siar, M.M., Roghanian, E., 2022. Sustainable, resilient and responsive mixed supply chain network design under hybrid uncertainty with considering COVID-19 pandemic disruption. *Sustain. Prod. Consumpt.* 30, 278–300.
- Vali-Siar, M.M., Roghanian, E., 2022. Resilient mixed open and closed-loop supply chain network design under operational and disruption risks considering competition: a case study. *Comput. Ind. Eng.* 172, 108513.
- Verbelen, R., 2013. Phase-type distributions & mixtures of Erlangs. *University of Leuven Faculty of Science Leuven Statistics Research Centre*, 126.
- Xu, Z., Saleh, J.H., 2021. Machine learning for reliability engineering and safety applications: Review of current status and future opportunities. *Reliab. Eng. Syst. Saf.*, 107530
- Yang, M., Lim, M.K., Qu, Y., Ni, D., Xiao, Z., 2023. Supply chain risk management with machine learning technology: a literature review and future research directions. *Comput. Ind. Eng.* 175, 108859.
- Yilmaz, Ö.F., Guan, Y., Yilmaz, B.G., 2025. A hybrid risk management framework for resilient medical supply chain design in the post-pandemic era. *J. Clean. Prod.*, 144854
- Zamani, E.D., Smyth, C., Gupta, S., Dennehy, D., 2023. Artificial intelligence and big data analytics for supply chain resilience: a systematic literature review. *Ann. Oper. Res.* 327 (2), 605–632.
- Zeynali, F.R., Parvin, M., ForouzeshNejad, A.A., Jeyzanibrahimzade, E., Ghanavati-Nejad, M., Tajally, A., 2025. A heuristic-based multi-stage machine learning-based model to design a sustainable, resilient, and agile reverse corn supply chain by considering third-party recycling. *Appl. Soft Comput.*, 113042
- Zheng, H., Ye, X., Chen, R., Chen, Y., 2025. Resilient high-end equipment manufacturing supply chain design with irreplaceable suppliers: an IFTOPSIS-MOMIP hybrid model. *Transp. Res. Part E: Logist. Transp. Res.* 195, 103914.