JOURNAL OF THE OPERATIONAL RESEARCH SOCIETY
2026, VOL. 77, NO. 6, 1453–1480
https://doi.org/10.1080/01605682.2025.2535713

RESEARCH ARTICLE

A novel near-to-expiry waste reduction model with smoothing ordering policy and retail discounting for food supply chains

Atieh Jafarian[a], Thi Le Hoa Vo[a], Madjid Tavana[b,c] and Emmanuelle Fromont[a]

[a]Supply Chain Management, University of Rennes, Rennes, France; [b]Business Systems and Analytics Department, Distinguished Chair of Business Analytics, La Salle University, Philadelphia, PA, USA; [c]Business Information Systems Department, University of Paderborn, Paderborn, Germany

ABSTRACT
This study presents a novel model for optimizing costs and reducing food waste in near-to-expiry (NTE) food supply chain (FSC) networks. The proposed approach involves direct product dispatch from manufacturers to distribution centers (DCs) and retailers through vehicle routing. The model incorporates a smoothing ordering policy and retail discounting to minimize waste by aligning purchasing decisions with real-time customer demand. A new metaheuristic, smart variable neighborhood search (SVNS) algorithm, is developed to address this problem. The SVNS algorithm dynamically selects neighborhood structures (NSs) to improve computational efficiency and solution quality. The performance of SVNS is evaluated against traditional benchmark methods through various experimental scenarios. The results show that the SVNS algorithm outperforms conventional benchmarks for managing near-expiration products.

ARTICLE HISTORY
Received 1 October 2024
Accepted 11 July 2025

KEYWORDS
Optimization; smart variable neighborhood search; smoothing ordering policy; food supply chain network; urban distribution network; perishable products

1. Introduction

Food waste is a global challenge with significant environmental, social, and economic implications. According to FAO (2023) and UNEP Food Waste Index Report (2021), around 14 % of the world's food is lost and wasted after it is harvested (before reaching retailers), and 17% of our food ends up being wasted in retail and by consumers. Ultimately, 1.3 billion tons of food are wasted yearly (Liu et al., 2021), with nearly 89 million tons belonging to the European Union (EU). The new EU law in 2018 requires member states to reduce food waste throughout the food supply chain (FSC) networks and track the level of waste to achieve Sustainable Development Goal 12.3 by reducing process and manufacturing waste by 10% and retail and consumption waste by 30% by 2030. Some research focused on reducing waste in FSC networks considers near-to-expiry (NTE) products. NTE foods, such as vegetables, fruits, and milk, are highly perishable and deteriorate quickly. In practice, they account for a significant share of food waste, primarily due to consumers' preference for fresh products and the limited shelf life that prevents long-term storage.

One of the most critical concerns of supply chains for NTE food products is customers' sensitivity to the aesthetic aspects of these products. For example, many consumers reject fruits and vegetables with minor imperfections, such as bananas with small brown spots or slightly misshapen apples. This sensitivity can lead to significant food wastage, especially at retail, where appearance is crucial in purchasing customers' decisions. They require sensitive storage conditions due to their perishable nature, their value decreases over time, and their usage period is very short (Lundqvist, 2009). Thus, this research focuses on NTE food waste reduction in the FSCs, emphasizing the retailers' role.

According to Kowalska and Manning (2021), 40% of the total waste is associated with FSC. A significant part of perishable food is related to retailers (Devin & Richards, 2018). Wastage primarily happens at the end of the supply chain, as the final product may become damaged or deteriorate during this stage (Liljestrand, 2017). The most common causes of food waste at retailers are irregular demand, inefficient storage (Teller et al., 2018), improper storage conditions (Buzby & Hyman, 2012), and overstocking (Buisman et al., 2019). Thus, accurate ordering policy and dynamic pricing can prevent food waste in retailers (Cicatiello et al., 2020). Implementing an effective ordering policy can help reduce food waste and increase profitability and customer satisfaction in retail centers (Estrada-Moreno et al., 2019).

Also, one strategy to address this issue involves discounting such products to make them more appealing to cost-conscious consumers (Cicatiello et al., 2019). Thus, discount as a dynamic pricing policy effectively convinces consumers to buy less favorable products

CONTACT Madjid Tavana tavana@lasalle.edu Business Systems and Analytics Department, Distinguished Chair of Business Analytics, La Salle University, Philadelphia, PA 19141, USA

and reduce food waste (Buisman et al., 2019). However, discounting should be used carefully to avoid creating a culture of overconsumption (Riesenegger & Hübner, 2022). Accordingly, discounting can be a valuable tool to reduce food waste in retailers, but it should be combined with other strategies, such as effective ordering policies and financial assistance to food banks and charities (Buisman et al., 2019). This article aims to reduce NTE food waste in FSCs, focusing on retail processes, including smoothing ordering policy and discounting. In this problem, retailers receive uncertain demand and order it based on forecasted demand, wasted products, and lost sales in previous periods. Retailers can also offer discounts to encourage customers to buy NTE foods.

While product ordering and discounting strategies play a pivotal role in reducing food waste, transportation planning is just as crucial for improving the efficiency and sustainability of FSCs. Research such as ensuring the timely delivery of products in the supply chain logistics (Davoudi et al., 2024), implementing logistics solutions at different stages of the supply chain, or integrating different activities and coordination mechanisms (Liljestrand, 2017), investigating the transportation problem to minimize the total (Mogale et al., 2017), reducing the environmental impacts of transportation (Shi et al., 2023), and integrating replenishment policies and transportation planning for reducing environmental impacts (Rout et al., 2021) are some of the recent advances in this field. One real-world example demonstrating the impact of integrated transportation planning and ordering strategies in reducing food waste and transportation costs is Tesco. One of the largest supermarket chains in the UK, Tesco, implemented a demand-driven ordering system and route optimization to enhance its FSC efficiency. By leveraging predictive analytics, Tesco accurately forecasts demand,[1] reduces overstocking by 25%, and minimizes waste from food products. Additionally, in partnership with Paragon, it was able to reduce transportation costs by 15% by integrating dynamic routing for delivery.[2] Thus, Tesco lowered its environmental impact and improved its profit margins by 5%, demonstrating that synchronized ordering and logistics can significantly enhance sustainability and cost-effectiveness in FSCs.

We develop a multi-product, multi-depot vehicle routing model with pickup and delivery. The proposed model aims to minimize the total costs across the network. The proposed problem is formulated as a stochastic mixed-integer programming model for vehicle routing within FSC networks, incorporating discounting strategies and a smoothing ordering policy. Since this problem integrates two NP-hard components – specifically, SCN design and the vehicle routing problem (VRP) – it is inherently classified as NP-hard (Govindan et al., 2014). Thus, we developed a novel meta-heuristic approach to solve this problem.

This study introduces several key contributions that distinguish it from previous studies. First, it integrates a smoothing ordering policy with discounting strategies at the retail level to minimize food waste and reduce total supply chain costs. The proposed ordering policy dynamically adjusts future orders based on past lost sales and waste, enhancing inventory efficiency. Second, it highlights the interconnectedness of logistics and retail operations – optimizing ordering and discounting mitigates food waste and lowers transportation costs in forward and reverse logistics. Also, the study considers a closed-loop supply chain, where wasted food is collected from retailers and returned for centralized disposal, thereby reducing environmental and health risks associated with abandoned spoiled products in urban areas. Lastly, it develops a smart variable neighborhood search (SVNS) algorithm tailored to the FSC design problem. This algorithm leverages ordinary neighborhood structures (NSs) as a general operator and directed NSs specifically designed for FSC design. This integration enhances exploration and exploitation, improving computational efficiency and algorithm performance.

The remainder of the article is organized as follows. Section 2 begins with a comprehensive literature review. Section 3 presents the problem description. Section 4 demonstrates the structure of the proposed algorithms. Section 5 presents the validation of the solution method. Sections 6 and 7 present the case study, its results, and its discussion. Finally, Section 8 concludes with our conclusions and future directions.

2. Literature review

This study builds upon three key streams in the literature, each of which contributes to the research novelty and impact. We explore these streams in detail and highlight our contributions within each domain.

2.1. Integrated ordering and discounting policies

As previously highlighted, retailers are pivotal in reducing food waste within the supply chain. Through various strategies, they can contribute to overall waste reduction efforts. Teller et al. (2018), Guritno et al. (2015), Qiu et al. (2019), and Wang et al. (2019) point out the effect of inventory management in reducing overstocking and food waste. Devin and Richards (2018) focus on food storage conditions that minimize food waste related to aesthetic aspects. Govindan et al. (2014) point out the effect of optimized inventory levels on retailers' food waste reduction, while other studies focus on the impact of replenishment policy on perishable products (Bottani et al., 2014; Somkun, 2017; Yang et al., 2017; Kiil et al., 2018). In addition, some studies, such as Chung and Li (2014), Adenso-Díaz et al. (2017), Buisman et al. (2019), Chen et al. (2019), Kayikci et al. (2022), and Dey et al. (2024) investigate the impact of the pricing strategies or discounting and dynamic pricing of perishable products in a retailer. However, few studies, such as Chintapalli (2015) and Lysova et al. (2024) have focused on integrating replenishment policy and discounting perishable items to minimize waste. Also, they implemented a blend of predictive and adaptive inventory replenishment policies, enabling dynamic inventory adjustments and effectively reducing waste. Integrating an ordering policy with a discounting strategy enhances inventory efficiency, reduces overstocking and food waste, and lowers total supply chain costs while improving demand responsiveness.

This study introduces an integrated approach that combines a smoothing ordering policy with discounting strategies at the retail level to minimize food waste and optimize total supply chain costs. The smoothing ordering policy dynamically adjusts future orders by considering past lost sales and unsold inventory, ensuring that retailers order more accurately in response to demand fluctuations. This approach reduces overstocking, prevents unnecessary waste, and enhances overall supply chain efficiency by aligning inventory levels with real-time sales patterns. Simultaneously, the strategic use of discounting incentivizes customers to use NTE products, further minimizing waste at the retail level while maintaining profitability. The study presents a practical and effective solution for balancing demand responsiveness and waste reduction in FSCs through this dual mechanism.

2.2. Waste in food supply chain networks

Several studies have focused on NTE products, introducing innovative distribution policies and models to accommodate their short shelf life. These approaches aim to align orders with demand fluctuations while reducing food waste across the supply chain (Zanoni & Zavanella, 2012). Some studies focus on perishability in the distribution process before delivery to retailers. For example, Tarantilis and Kiranoudis (2001) study the distribution of fresh milk to a set of retailers before the perishability time. Other studies have focused on perishability at retailers; for instance, Xu et al. (2020) provide a decision-support framework for reducing waste in fresh apple supply chains. Lütke Entrup et al. (2005) integrate product shelf life issues and production planning of yogurt. Several studies have examined the environmental impact of food SCNs, addressing key issues such as greenhouse gas emissions and food waste. These studies emphasize the role of sustainable logistics, eco-friendly transportation, and waste reduction strategies in mitigating environmental damage. Martins and Pato (2024) examine the impact of food waste reduction on CO_2 emissions and total costs. Their study also demonstrates how the return of wasted food for disposal or recycling increases transportation costs in reverse logistics.

Several studies have proposed circular economy models, closed-loop supply chains, and advanced waste management techniques to minimize the ecological footprint of food distribution and retail operations. Wang et al. (2022) emphasize the importance of incorporating environmental considerations at all product life cycle stages to achieve sustainability and regulatory compliance. Kazimierowicz (2018) highlights that a centralized disposal process helps prevent the improper disposal of spoiled items at the retail or city level, reducing the risk of environmental contamination and public health hazards. Also, Sarvary and Padmanabhan (2001) emphasize the strategic importance and environmental impacts of the return policy for wasted products in the manufacturer-retailer relationship. They highlight that analyzing market conditions through wasted products can provide valuable insights, enabling manufacturers to optimize the FSC more effectively. Guide Jr and Vanwassenhove (2001) show that the secure return of expired products to manufacturers addresses multiple concerns, including minimizing environmental impact and ensuring proper waste management.

Despite the existing research on food redistribution and waste collection, limited attention has been paid to the integrated design of delivery and pickup operations in the FSC. Thus, this study aims to reduce food waste by implementing an ordering policy and discounting strategy, consequently increasing transportation costs in reverse logistics. Additionally, we use a VRP with pickup and delivery, ensuring the collection of wasted food from retailers and its return to manufacturers for centralized disposal. Through this approach, the model addresses environmental concerns by minimizing food waste, ensuring proper disposal, and reducing overall costs, waste, and health risks associated with abandoned spoiled products in urban areas.

2.3. Developing smart variable neighborhood search

SCN design problems are classified as NP-hard, meaning that exact solution methods require exponential computational time, making them impractical for large-scale instances. As a result, heuristic and metaheuristic approaches are widely adopted to tackle these challenges (Mostajabdaveh et al., 2019). While heuristic methods offer high computational efficiency, they often lack accuracy in finding high-quality solutions (Wang, 2023). In contrast, meta-heuristic approaches balance computational efficiency and solution quality by effectively navigating vast and complex solution spaces. Therefore, meta-heuristics are generally preferred for large-scale SCN design problems, as they can provide near-optimal solutions within a reasonable computational time (Govindan et al., 2014). Thus, methods such as genetic algorithms (GAs), particle swarm optimization (PSO), and tabu search (TS) have been extensively explored in the literature to optimize facility location, inventory management, and transportation decisions (Sadeghi et al., 2014; Soleimani & Kannan, 2015; Alavi et al., 2016; Mousavi et al., 2017; Karami, 2022; Zhou et al., 2023). However, the variable neighborhood search (VNS) algorithm has gained significant popularity due to its flexibility in exploring different NSs, ability to escape local optima efficiently, and relatively simple implementation compared to other metaheuristics. The extensive use of VNS in SCN design stems from its capability to provide high-quality solutions within reasonable computational times, making it a preferred choice for static and dynamic network design.

VNS algorithms have evolved into classifications, each incorporating unique enhancements to improve solution quality and computational efficiency as follows:

i. *Adaptive VNS* refers to any VNS method that dynamically modifies not only its parameters but also its structural components, such as the selection, ordering, or type of NSs, throughout the search process. This broader definition aligns with the perspective of Hansen et al. (2017), who emphasize that adaptive VNS includes both parameter tuning and structural adaptation to guide the search more intelligently and responsively. Thus, the input parameters adjust themselves dynamically and iteratively based on the performance of the algorithm, allowing the search strategy to adapt to the problem characteristics. Furthermore, NSs are dynamically selected and sorted based on their past effectiveness, and real-time feedback is used to influence future decisions. These adaptations enable the algorithm to ensure a balance between exploration and exploitation and improve the convergence speed. Also, Nunes Bezerra et al. (2019), Hà et al. (2020), and Sze et al. (2024) propose an adaptive VNS approach for the VRP that tunes the input parameters iteratively based on algorithm performance. Schneider et al. (2015) developed an adaptive VNS algorithm that adjusts the weights of NSs iteratively for a VRP requiring stops at specific facilities for continued service. Karakostas and Sifaleras (2022) introduced the double-adaptive GVNS with two basic adaptive mechanisms: adjusting the shaking intensity and reordering NSs based on past performance for solving the traveling salesman problem. Building upon their earlier work, Karakostas and Sifaleras (2024) extended the double-adaptive GVNS by proposing a more advanced version that integrates knowledge-based adaptation mechanisms. The algorithm adjusts the intensity of shaking, reorders NSs, and leverages accumulated performance data to guide the search process more intelligently, thereby improving efficiency. Also, Liu et al. (2023) propose a dual adaptive mechanism that dynamically adjusts the selection of NSs for the shaking procedure and local search based on their past performance. The mechanism utilizes roulette wheel selection (RWS) to choose NSs, where weights are updated based on their effectiveness in improving solutions over past iterations, to prioritize the most promising search strategies while maintaining diversification.

ii. *Intelligent VNS* incorporates artificial intelligence (AI) techniques such as machine learning to optimize real-time parameter selection. By leveraging AI-driven models, Intelligent VNS can predict the most effective parameter settings at each iteration, reducing the need for manual tuning and enhancing overall search performance (Chen et al., 2020; Kamran et al., 2023; Alrashidi & Al Ghamdi, 2024).

iii. *Smart VNS* (SVNS) introduces a directive NS that refines solutions heuristically. Directive NSs improve the performance of VNS by guiding the search process toward more promising solution regions, reducing unnecessary exploration of low-quality neighborhoods (Bezerra et al., 2023). Unlike traditional random or systematic neighborhood changes, directive structures prioritize moves that are more likely to improve the solution based on heuristic rules or problem-specific insights. This targeted approach enhances exploration efficiency, accelerates convergence, and reduces computational time while maintaining solution quality (Govindan et al., 2019).

Our proposed algorithm enhances neighborhood selection based on problem characteristics, optimizes neighborhood switching methods, and restricts search spaces to the most promising regions, accelerating the solving process. This algorithm improves the efficiency of the original algorithm by using innovative strategies without using adaptive learning or intelligent methods.

The proposed algorithm uses an explicitly designed search structure for this problem. We integrate ordinary and directed NSs to prevent premature convergence to a local optimum. While directed neighborhoods guide the search efficiently, ordinary NSs expand the exploration space, ensuring a broader search.

3. Problem description

As shown in Figure 1, the desired supply chain has three stages: manufacturers, distribution centers (DCs), and retailers. The manufacturers produce products and send them to the DCs. The products are dispatched from manufacturers to DCs served by the 1st-echelon transportation. The 1st-echelon transportation considers direct shipments that involve handling large quantities of homogeneous cargo sent in the standard pallet. Each 1st-echelon vehicle travels to a DC and returns the wasted products to the original manufacturer for centralized disposal. As the products considered in this study are NTE food products, the storage process is not considered in any layer of the supply chain. The main operation in DC involves receiving, sorting, and dispatching goods. DCs efficiently manage and fulfill orders and ensure timely delivery to retail locations.

A DC receives orders from all related retailers. The total order from the retailers to a DC equals that from a DC to a manufacturer. The routing policy is used in the 2nd-echelon transportation, as the shipments to retailers are classified as less-than-truckload. In this transportation echelon, the vehicles start from a DC, navigate a prescribed route to serve designated retailers, and return to the same DC. DCs also collect expired products from retailers from the previous period.

Retailers often discard deteriorated products in dumpsters across the city, exacerbating environmental pollution due to improper disposal processes. A centralized disposal system managed by manufacturers can serve as an effective solution for handling waste responsibly. Most manufacturers use recycling or incineration to dispose of deteriorated products safely. In this study, food waste occurs at retailers due to a lack of consumer purchases. This waste imposes three types of costs on the supply chain: the cost of the unsold product, the transportation cost for returning the wasted items, and the cost of disposal. Accordingly, trucks transport new products to retailers and concurrently collect wasted products for return to the DCs. While an operational cost is assigned for handling and collecting wasted products at the retailers, the transportation cost for returning these products to the DCs is not considered. This is because the trucks must return to their origin (the DCs) regardless of whether they carry wasted products, making the return journey an inherent part of the delivery route. Therefore, the cost of returning wasted products is included in the overall delivery route cost. The model avoids double-

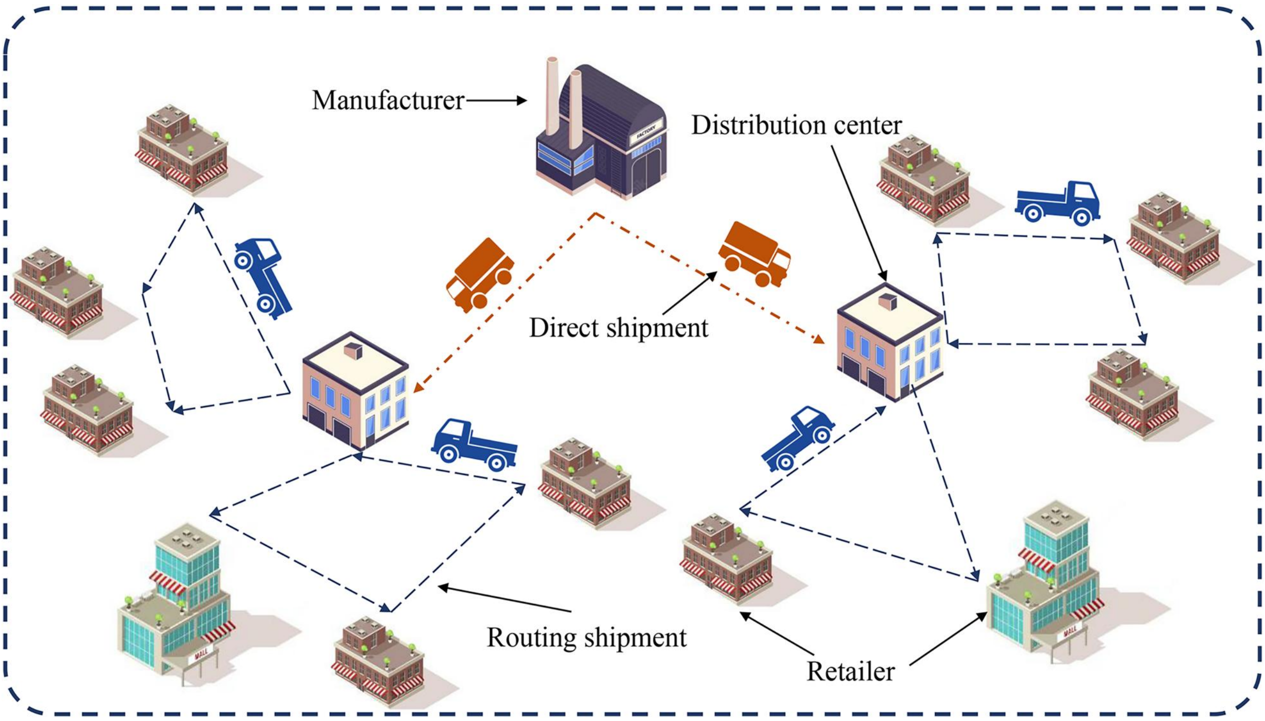

Figure 1. The food distribution network problem.

counting transportation expenses while maintaining an efficient and cost-effective supply chain structure by leveraging this backhauling approach, where vehicles utilize their return trips for waste collection. The retailers contribute to food waste reduction by a replenishment policy known as the smoothing ordering policy and a discounting policy described as follows:

3.1. Smoothing ordering

In the inventory part of the problem, we suggest a smoothing ordering policy that simultaneously reduces food waste and lost sales. As a result, retailers often rely on past demand to predict future orders, which leads to the implementation of replenishment strategies known as the order-up-to (OUT) policy (Cannella & Ciancimino, 2010). Accordingly, the amount of replenishment based on forecast orders is calculated by the retailers as follows:

$$O_{rgp} = \hat{d}_{rgp} - \beta_{rg} s_{rgp-1} + \delta_{rg} l_{rgp-1} \tag{1}$$

where $\hat{d}_{rgp}$ is the forecasted demand of retailer $r \in R$ for the product $g \in G$ in the period $p \in P$, is calculated by a single exponential smoothing method (Jafarian et al., 2019). Also, s_{rgp} and l_{rgp} represent the quantities of unsold and lost sales of the product, respectively, incurred by the retailer $r \in R$ during period $p \in P$. β_{rg} and δ_{rg} are constants used to account for the impact of deteriorated products and lost sales from the previous period, respectively, in forecasting the demand for product $g \in G$ for period $p \in P$, respectively. β_{rg} and δ_{rg} are correction factors that adjust the predicted demand based on the quantities of deteriorated products and lost sales from the previous period.

3.2. Discounting

A retailer's price discount, particularly on products selling below expected volumes, is a tactical response to stimulate demand and clear excess remaining products. By reducing the price of underperforming products, the retailer seeks to attract more customers and incentivize purchases, leveraging the basic economic principle of price elasticity of demand; as the price level decreases, the quantity demanded generally increases. The decision to discount prices during the day allows the retailer to adjust to market conditions and consumer behavior dynamically, maximizing sales opportunities and minimizing potential losses from unsold products. We use the concept of discounting proposed by Cachon and Swinney (2009).

Thus, each period is divided into two parts $p = p1 + p2$. In the first part of a period, the retailer sells the Product $g \in G$ at a primary price (ρ_g). In this situation, the demand is divided into two parts $d_{rgp} = d_{rgp1} + d_{rgp2}$, assuming that the demand is distributed equally and symmetrically, i.e., $d_{rgp1} = d_{rgp2}$. As soon as the second part is started, if the retailer detects that the amount of demand is lower than half of the remaining products, i.e., $d_{rgp1} \leq O_{rgp-1}/2$. It can decide to offer a discount and sell all the remaining products. In this regard, customers are classified into two groups: (1) *Normal customers*, those who have regular purchases and (2) *Bargain hunters*, those who buy the product at a discount. It is assumed that the number of bargain-hunter customers is infinite; in this case, if the price is discounted, they will buy all the remaining products (Cachon & Swinney, 2009). Although strategic customers may change their behavior because of the discount and transfer their demand to the discounted part (Cachon & Swinney, 2009), we ignore them. Also, in periods where there is a discount, a lost sale is not considered. Retailers can decide on a product discount at a defined level, so it is a binary decision variable. After the discount, the demand during the second period is equal to the regular customers, i.e., $d_{rgp1} = d_{rgp2}$ plus the bargain hunters attracted by the discount, i.e., $O_{rgp-1} - d_{rgp1}$. It should be noted that in the second part, both groups buy the products at a discount price, i.e., $\rho_g(1 - \vartheta_{rg})$.

Before formulating the problem, the following assumptions are introduced:

- The number, position, and capacity of retailers and DCs are known.
- There are vehicles with specific capacities.
- Manufacturers and DCs have limited capacity.
- Each retailer is visited at most once in each period.
- Each vehicle is connected to only one DC.
- The cost of travel/distance unit is known.
- First-echelon trips must begin with manufacturers and end at DCs, and vice versa for wasted goods.
- Second-echelon trips must begin/ end at the same DCs.
- The manufacturer cannot ship the product directly to retailers.

The following notations are used to formulate the problem:

Indices	Sets
k	Set of vehicle types $k \in \{1, 2, ..., K\}$
m	Set of manufacturers $m \in \{1, 2, ..., M\}$
d	Set of distribution centers $d \in \{1, 2, ..., D\}$
r	Set of retailers $r \in \{1, 2, ..., R\}$
p	Set of period time $p \in \{1, 2, ..., P\}$
ξ	Set of demand scenarios $\xi \in \{1, 2, ..., \Xi\}$
Parameters	**Description**
$d_{rgp}(\xi)$	The demand for the Product g received by the retailer r in the period p in the scenario ξ, including $d_{rgp1}(\xi), d_{rgp2}(\xi)$
α	The smoothing parameter in simple exponential smoothing $(0 \leq \alpha \leq 1)$
Q_d Q_m Q	Capacity of the DC, and manufacture, and truck
Q_{rg}	Capacity of retailer r for product g
O_{rgp}	Amount of the product g ordered by the retailer $r \in R$ in the period $p \in P$
TC_{md}	Transport cost in the first transportation echelon
C_{ij}	Average cost of traveling from node i to node j in the 2nd transportation echelon $(i, j \in A, i \neq j)$
OC_d	Operation cost in a distribution center d
RC_{dg}	Returning cost includes the operational cost of wasted products in DCs for receiving and repacking.
ρ_g	Price of product g
DC_{mg}	Disposal cost of product g at manufacturer m
θ_g	Cost of lost sales product g
ϑ_{rg}	Discount level can be considered by the retailers r for the product g
N	The set of nodes consists of $\{D \cup R\}$
S	The subset of retailers that a truck can visit on a route $(S \in R)$
M	A big number

The decision variables are defined as follows:

Decision Variable	Description
Binary variables	
$x^k_{ijp}(\xi)$	1 if vehicle k traverses arc $(i, j) \in N$ in the period p in the scenario ξ, 0 otherwise
$z^k_{ip}(\xi)$	1 if vehicle k visits the retailer r in the period p in the scenario ξ, 0 otherwise
$e_{rgp}(\xi)$	1 if a retailer decides to discount the product g in the period p in the scenario ξ, 0 otherwise
$\nu_{mdg}(\xi)$	1 if the DC d is assigned to manufacture m to prepare *the* product g in the scenario ξ, 0 otherwise
$\kappa_{rgp}(\xi)$	1 if the retailer r is allowed to discount the price of the product g in the period p in the scenario ξ, 0 otherwise
Continuous variables	
$y^k_{rgp}(\xi)$	Number of products delivered to the retailer r by vehicle k in period p in scenario ξ
$u^k_{rgp}(\xi)$	Number of wasted products collected by vehicle k from retailer r in period p in scenario ξ
$\beta_{rg}(\xi)$	Constant correcting factor, decided by the retailer r that considers the effect of wasted products of the previous period in ordering the product g in the scenario ξ
$\delta_{rg}(\xi)$	Constant correcting factor, decided by the retailer r that considers the lost sales of the previous period in ordering product g in scenario ξ
$b^k_{rgp}(\xi)$	Used capacity of vehicle k in the period p in scenario ξ
$h_{dgp}(\xi)$	Number of products operated at DC d in period p in scenario ξ
$w_{dgp}(\xi)$	Number of the wasted products received at DC d in the period p in scenario ξ
l_{rgp}	Number of lost sales for the product g in retailer r in period p
s_{rgp}	Amount of unsold product g by the retailer r in period p

The uncertainty of retailers' demand is considered by formulating a scenario-based stochastic programming model. The uncertainty of the request is illustrated by assuming a number of scenarios, $\{1, 2, ..., |\xi|\}$, each with a possibility $P(\xi)$, that probabilities satisfy the condition of $\sum_{\xi \in \Xi} P(\xi) = 1$. The objective function aims to minimize the total costs within the FSC network, covering the following cost components:

- ***Transportation costs:*** These expenses include transportation costs for both forward and reverse logistics: in the first echelon, they include the cost of delivering products from manufacturers to DCs and returning wasted products from DCs back to manufacturers; in the second echelon, they encompass the cost of distributing products from DCs to retailers.

$$\sum_{m\in M}\sum_{d\in D}TC_{md}\sum_{g\in G}\nu_{mdg}(h_{dgp}(\xi)+w_{dgp}(\xi)) + \sum_{i\in N}\sum_{j\in N,\, i\neq j}\sum_{k\in K}C_{ij}x^k_{ijp}(\xi) \tag{2}$$

- ***Operation costs at DCs***: These costs refer to the expenses incurred during DC activities and include the costs associated with receiving and sorting new products delivered from manufacturers and collecting, receiving, and sorting wasted products returned from retailers.

$$\sum_{d\in D}OC_d\sum_{g\in G}(h_{dgp}(\xi)+w_{dgp}(\xi)) + \sum_{d\in D}\sum_{g\in G}RC_{dg}w_{dgp}(\xi) \tag{3}$$

- ***Retailing costs***: These costs include three types of expenses related to sales and inventory management. *i*) Discounting costs represents lost profit from price reductions on NTE products. *ii*) Lost sales costs occur when stockouts prevent customers from purchasing desired products, leading to missed revenue opportunities. *iii*) Unsold product costs result from over-ordering, causing excess inventory that remains unsold and ultimately becomes waste.

$$\sum_{r\in R}\sum_{g\in G}(O_{rgp}-d_{rgp-1})\rho_g\vartheta_{rg}\ e_{rgp}(\xi) + \sum_{r\in R}\sum_{g\in G}l_{rgp}\theta_g(\xi) + \sum_{r\in R}\sum_{g\in G}s_{rgp}\rho_g(\xi) \tag{4}$$

- ***Disposal costs***: These costs cover the expenses associated with disposing of waste products, including proper waste management processes such as recycling or environmentally safe disposal methods.

$$\sum_{m\in M}\sum_{g\in G}DC_{mg}\sum_{d\in D}\nu_{dgp}w_{dgp}(\xi)) \tag{5}$$

The mathematical model of the problem is presented as follows:

$$\begin{aligned}\min\ OBJ = &\sum_{p\in P}(\sum_{m\in M}\sum_{d\in D}TC_{md}\sum_{g\in G}\nu_{mdg}(h_{dgp}(\xi)+w_{dgp}(\xi)) \\ &+\sum_{i\in N}\sum_{j\in N,\, i\neq j}\sum_{k\in K}C_{ij}x^k_{ijp}(\xi) \\ &+\sum_{d\in D}\sum_{g\in G}RC_{dg}w_{dgp}(\xi)+\sum_{d\in D}OC_d\sum_{g\in G}(h_{dgp}(\xi)+w_{dgp}(\xi)) \\ &+\sum_{r\in R}\sum_{g\in G}(O_{rgp}-d_{rgp-1})\rho_g\vartheta_{rg}\ e_{rgp}(\xi)+\sum_{r\in R}\sum_{g\in G}l_{rgp}\theta_g(\xi) \\ &+\sum_{r\in R}\sum_{g\in G}s_{rgp}\rho_g(\xi)+\sum_{m\in M}\sum_{g\in G}DC_{mg}\sum_{d\in D}\nu_{dgp}w_{dgp}(\xi)))\end{aligned} \tag{6}$$

The constraints of the model are as follows:

Forward flow constraints:

$$\sum_{i\in N}\sum_{k\in K}x^k_{ijp}(\xi)\leq 1 \quad \forall j\in R, p\in P, \xi\in\Xi \tag{7}$$

$$\sum_{j\in N}x^k_{ijp}(\xi)+\sum_{j\in N}x^k_{jip}(\xi)=2z^k_{ip}(\xi) \quad \forall i\in N, k\in K, p\in P, \xi\in\Xi \tag{8}$$

$$\sum_{i\in S}\sum_{j\in S,\, i\neq j}x^k_{ijp}(\xi)=|S|-1 \quad \forall k\in K, S\in R, |S|\geq 2, p\in P, \xi\in\Xi \tag{9}$$

Constraints (7) ensure each retailer can be visited a maximum of once in a period. Constraints (8) impose the flow conservation on each retailer in each period and guarantee that a vehicle returns to its DC of origin. Constraints (9) are sub-tour elimination constraints and ensure that each retailer is visited once in each period.

Ordering and discounting constraints:

$$O_{rgp}=\hat{d}_{rgp}-\beta_{rg}s_{rgp-1}+\delta_{rg}l_{rgp-1} \quad \forall r\in R, g\in G, p\in P \tag{10}$$

$$\hat{d}_{rgp}=\alpha d_{rgp-1}+(1-\alpha)\hat{d}_{rgp-1} \quad \forall r\in R, g\in G, p\in P\forall r\in R, g\in G, p\in P \tag{11}$$

$$s_{rgp}=\sum_{k\in K}y^k_{rgp}(\xi)-d_{rgp}-(O_{rgp-1}-d_{rgp})e_{rgp}(\xi)+l_{rgp} \quad \forall r\in R, g\in G, p\in P, \xi\in\Xi \tag{12}$$

$$\sum_{k\in K}y^k_{rgp}(\xi)=O_{rgp-1} \quad \forall r\in R, g\in G, p\in P, \xi\in\Xi \tag{13}$$

$$\sum_{k\in K}u^k_{rgp}(\xi)=s_{rgp-1} \quad \forall r\in R, g\in G, p\in P, \xi\in\Xi \tag{14}$$

$$d_{rgp1}-(O_{rgp-1}/2)\leq M\times(1-\kappa_{rgp}(\xi)) \quad \forall r\in R, g\in G, p\in P, \xi\in\Xi \tag{15}$$

$$d_{rgp1}-(O_{rgp-1}/2)\geq -M\times\kappa_{rgp}(\xi) \quad \forall r\in R, g\in G, p\in P, \xi\in\Xi \tag{16}$$

$$e_{rgp}(\xi)\leq\kappa_{rgp}(\xi) \quad \forall r\in R, g\in G, p\in P, \xi\in\Xi \tag{17}$$

Constraints (10) calculate the order quantity based on forecasted demand and the degree of importance of lost sales and wasted products. Constraints (11) predict the market for a product by simple exponential smoothing. Constraints (12) calculate the inventory balance, stating that the unsold products should equal the received products minus received demand and that increased demand is created by a discount plus lost sales. Inequalities (13) impose that the product delivered to a retailer must be less than what was ordered in the previous period. Constraints (14) show that products returned from a retailer are equal to the prior period unsold products of that retailer in the last period. Constraints (15) and (16) impose that if the demand received in the first part of a period is more than the available product for selling in that part of the period, the retailer can consider a discount. The binary decision variable κ_{rgp} is defined as discount permission in the if-then conditions (18) and transformed into a linear equation in constraints (15)–(16). Furthermore, Constraints (17) show the permission of discounting for retailers, i.e., e_{rgp} must be zero if $\kappa_{rgp} = 0$, and e_{rgp} can be 0 or 1, if $\kappa_{rgp} = 1$.

$$\kappa_{rgp} = \begin{cases} 0 & \text{if } d_{rgp1} - (O_{rgp-1}/2) \geq 0 \\ 1 & \text{if } d_{rgp1} - (O_{rgp-1}/2) < 0 \end{cases} \quad \forall r \in R, g \in G, p \in P, \xi \in \Xi \tag{18}$$

Backward flow constraints:

$$b^k_{jgp}(\xi) = (b^k_{ijp} - y^k_{jgp} + u^k_{jgp})x^k_{ijp}(\xi) \quad \forall i \in N, j \in R, g \in G, k \in K, p \in P, \xi \in \Xi, \tag{19}$$

$$\sum_{k \in K}\left(\sum_{r \in R} x^k_{drp}(\xi)\right)\left(\sum_{r \in R} y^k_{rgp}(\xi)\right) = h_{dgp}(\xi) \quad \forall d \in D, g \in G, p \in P, \xi \in \Xi \tag{20}$$

$$\sum_{k \in K}\left(\sum_{r \in R} x^k_{rdp}(\xi)\right)\left(\sum_{r \in R} u^k_{rgp}(\xi)\right) = w_{dgp}(\xi) \quad \forall d \in D, g \in G, p \in P, \xi \in \Xi \tag{21}$$

$$\sum_{m \in M}\sum_{g \in G} v_{mdg}(\xi) = 1 \forall d \in D, \xi \in \Xi \tag{22}$$

Constraints (19) ensure that the vehicle capacity limits are respected during product deliveries and waste collection at each node. Constraints (20)–(21) calculate the products and returned waste products that should be operated in a DC. Constraints (22) impose that a DC can be assigned to a manufacturer.

Capacity constraints:

$$O_{rgp} \leq Q_{rg} \quad \forall r \in R, g \in G, p \in P, \xi \in \Xi \tag{23}$$

$$\sum_{r \in R}\sum_{g \in G} y^k_{rgp}(\xi) \leq Q \quad \forall k \in K, p \in P, \xi \in \Xi \tag{24}$$

$$\sum_{r \in R}\sum_{g \in G} u^k_{rgp}(\xi) \leq Q \quad \forall r \in R, k \in K, p \in P, \xi \in \Xi \tag{25}$$

$$\underset{t}{y^k_{rgp}} + u^k_{rgp} \leq M \times \sum_{i \in N} x^k_{irp}(\xi) \quad \forall r \in R, k \in K, g \in G, p \in P, \xi \in \Xi \tag{26}$$

$$\sum_{g \in G} b^k_{rgp} \leq Q \quad \forall r \in R, k \in K, p \in P, \xi \in \Xi \tag{27}$$

$$\sum_{g \in G} h_{dgp}(\xi) + w_{dgp}(\xi) \leq Q_d \quad \forall d \in D, p \in P, \xi \in \Xi \tag{28}$$

$$\sum_{d \in D}\sum_{g \in G} v_{mdg}(\xi)(h_{dgp} + w_{dgp}) \leq Q_m \quad \forall m \in M, p \in P, \xi \in \Xi \tag{29}$$

Constraints (23) require that the orders a retailer sends should be less than its ordering capacity. SCs require retailers to order less than their maximum ordering capacity to ensure a more stable and predictable flow of goods through the network and to reduce the risk of stockouts or unsold products. By limiting order sizes, DCs can better allocate resources, maintain consistent lead times, and ensure that products are available to a broader range of retailers. Constraints (24) and (25) guarantee that the capacity of vehicles should be observed when loading the products and unloading the waste products in the depot. Constraints (26) illustrate that if a vehicle does not visit a retailer, the product amount delivered/returned by the vehicle must be zero. Constraints (27) guarantee the vehicle's capacity while visiting a retailer. Constraints (28) and (29) impose that the DC's and manufacturer's capacities should be observed, respectively. Finally, integrality and non-negativity of variables are guaranteed by:

$$x^k_{ijp}(\xi), z^k_{ip}(\xi), e_{rgp}(\xi), \nu_{rgp}(\xi), \kappa_{rgp}(\xi) \in \{0, 1\} \tag{30}$$

$$y^k_{rgp}(\xi), u^k_{rgp}(\xi), \beta_{rg}(\xi), \delta_{rg}(\xi), b^k_{rgp}(\xi), h_{dgp}(\xi), w_{dgp}(\xi) \geq 0 \tag{31}$$

4. Proposed solution algorithm

The SCN design problem is classified as an NP-hard problem, indicating its high computational complexity. This complexity arises from integrating the VRP and ordering policy within the SCN model, making the formulated problem particularly challenging due to its large solution space and combinatorial nature. As the problem size increases, the solution time increases exponentially with exact methods (Woeginger, 2003). Thus, the metaheuristic/heuristic approaches are a suitable and promising option for obtaining high-quality solutions, especially for large-scale problems (Nguyen et al., 2012). Still, they are not the only available methods. Meta-heuristics provide significant advantages over traditional heuristics, particularly regarding flexibility, scalability, and the ability to find near-optimal solutions for complex optimization problems. Unlike simple heuristics, which are often problem-specific and may struggle to scale or adapt to changes in problem characteristics, meta-heuristics are designed to explore the solution space more comprehensively and efficiently. Their versatile search strategies enable them to handle various optimization problems, making them particularly effective for large-scale and complex solution spaces (Chopard et al., 2018). Therefore, the SVNS algorithm, an enhanced and redesigned version of the original VNS, is developed to address this problem effectively.

The SVNS algorithm differentiates itself from the original VNS through two innovative adaptations tailored for complex optimization problems like those encountered in food SCN integrated with VRP. First, SVNS introduces a novel policy for selecting NSs, prioritizing those that significantly reduce computational time without compromising the exploration depth. This approach contrasts with VNS's more generic neighborhood shifting mechanism, enabling SVNS to be more efficient and faster in converging towards optimal or near-optimal solutions. Second, the adoption of directed NSs in SVNS specifically addresses the unique challenges of the problem domain, such as routing and logistics in supply chains, by guiding the search process toward more promising areas of the solution space. These purposefully designed structures enhance the solution's quality while trimming the computation time, making SVNS a more targeted and effective meta-heuristic for tackling the intricacies of integrated supply chain and routing problems. In the following subsections, SVNS is presented comprehensively.

4.1. Encoding scheme and initialization

This article uses the Random Key (RK) technique in the SVNS algorithm. The RK technique enables meta-heuristic algorithms to be developed based on a continuous approach to solve this discrete problem (Chang et al., 2009). This feature provides the powerful exploration and optimization capabilities of continuous meta-heuristics to be harnessed for discrete problem domains, offering a flexible and versatile method for finding high-quality solutions to complex optimization challenges (Devika et al., 2014). The RK technique involves encoding the solution of a discrete problem as a vector of real numbers (primary solution), commonly referred to as "random keys." Each position in the vector corresponds to a component of the discrete problem's solution, and the numerical values are used to determine the order or selection of these components through a decoding process (parsed solution) (Govindan et al., 2015). The encoding of a solution for the proposed problem is schematically illustrated in Figure 2. The solution includes five sub-solutions. In the first sub-solution, DCs are assigned to manufacturers. A matrix with $|M \times D|$ elements, each having a uniform distribution, is generated, where RK is used to assign DCs to the manufacturer. Thus, in each column, the maximum element is selected. For example, the encoded solution $\{0.94, 0.69, 0.41, 0.08; 0.50, 0.42, 0.77, 0.98\}$ represents the parsed solution $\{1, 1, 0, 0; 0, 0, 1, 1\}$ indicating, that DCs 1 and 2 are assigned to Manufacturer 1, DCs 3, and 4 are assigned to Manufacturer 2.

In the second sub-solution, the routes of the vehicles are determined. A matrix with $|R + K - 1|$ elements each with $U(0, 1)$ is produced. The numbers are then sorted in descending order to indicate the order of retailers the vehicles visit. The last $K - 1$ elements act as the dividers of vehicles. For instance, imagine two vehicles that want to meet six retailers. A matrix with seven elements is produced. Then, the parsed solution represents that is served by vehicle 1 and is served by vehicle 2, and the fourth element is a divider. According to sub-solution 3, a matrix with K elements each with $U(1, DC)$ are generated. Then, the RK rounds numbers and demonstrates desired DCs. For example, the encoded solution $\{1.65, 1.09\}$, represents the parsed solution $\{2, 1\}$ indicating, that vehicle one that serves to R_4, R_2, R_5 is assigned to DC2, and vehicle two that services to R_1, R_6, R_3, assigned to DC1.

The fourth sub-solution shows correction constants of wasted products (β) and, correction constants of unsold products (δ) determined by each retailer for each product. A series of initial experiments showed that the randomness of these two should be set between 0.1 and 0.9. So at first, a random number between 0 and 1 is generated in a matrix with $|2 \times R|$ elements each with $U(0, 1)$ for

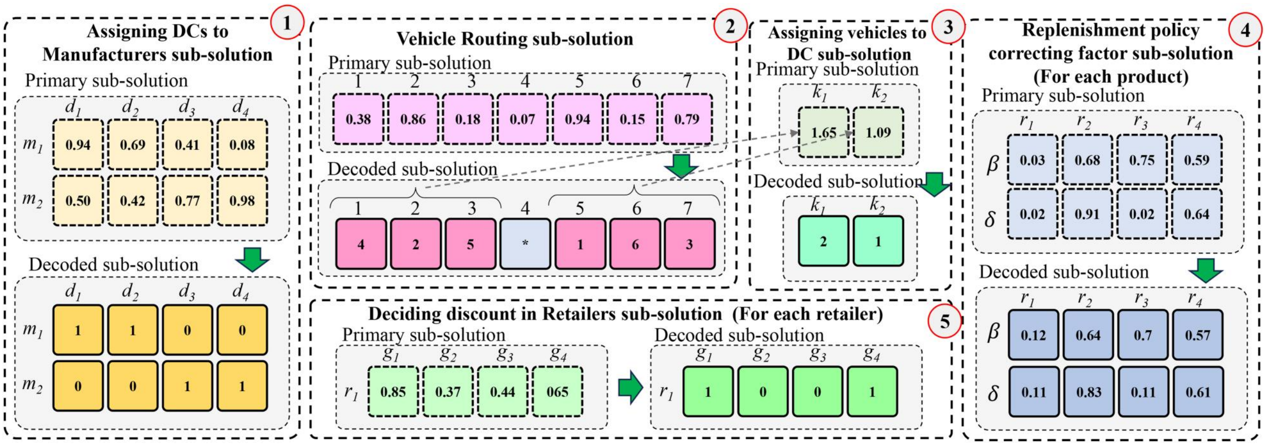

Figure 2. Initialization and encoding.

each product. A lower and upper limit is defined for β, and δ to change within this range i.e., $[\beta_{\min}, \beta_{\max}]$, and $[\delta_{\min}, \delta_{\max}]$. Thus, the RK is used to parse the solution as the primary solution is multiplied by $(\beta_{\max} - \beta_{\min}) + \beta_{\min}$ (same for δ). For example, the encoded solution $\{0.03, 0.68, 0.75, 0.59\}$, with $[\beta_{\min}, \beta_{\max}] = [0.1, 0.9]$ represents the parsed solution $\{0.12, 0.64, 0.7, 0.57\}$. The last sub-solution clears the discount determined by each retailer in each period. Thus, first a matrix with G elements, each with $\mathrm{U}(0, 1)$ is generated for each retailer. Then as the decision to discount is a binary variable, the continuous number is converted to discrete by the RK technique to represent the retailer's decision to apply a discount. For example, the encoded solution $\{0.85, 0.37, 0.44, 0.65\}$ represents the parsed solution $\{1, 0, 0, 1\}$ obtained by rounding the encoded solution. Therefore, the parsed solution reveals that the retailer considers the discount for the first and fourth products.

The initialization procedure in SVNS generates the initial *nPop* particles as a first population. Thus, the initial solutions are generated randomly within feasible space with an N-dimensional hyperspace. Then, the RK technique parses the primary solutions to the corresponding values of each particle in discrete areas. During this conversion, the parsed solutions produced by the RK method are automatically feasible, as the parsing process aligns the solutions with the defined problem constraints. Therefore, the RK technique eliminates the need for additional feasibility checks, as all parsed solutions inherently meet structural constraints such as route sequencing or assignment rules. However, a penalty value is introduced into the model to ensure adherence to capacity limitations. This penalty imposes a cost on solutions that violate capacity constraints, guiding the search process to remain within the feasible solution space while exploring potential solutions effectively.

4.2. SVNS algorithm

The VNS algorithm, introduced by Mladenović and Hansen (1997), is a metaheuristic method that systematically explores the solution space of an optimization problem by dynamically changing the NSs. The core idea behind VNS is that different NSs can reveal new local optima, allowing the algorithm to escape local minima and enhance the search for a global optimum (Hansen & Mladenovic, 2003). This method is particularly effective for large-scale problems where the solution space is complex, as it balances local intensification with global diversification. Also, it has been successfully used in many practical problems (Karakostas & Sifaleras, 2022), such as VRP (Bräysy, 2003) and SCN design problems (Eskandarpour et al., 2013). In general, VNS has been applied to many real-world problems and has provided better performance than other methods (Brimberg et al., 2023). For more related literature about VNS, readers are referred to Mladenović and Hansen (1997) and (Hansen et al., 2017). The VNS algorithm explores the solution space through systematic NS changes, starting from an initial solution and applying local modifications to enhance the objective value. In each iteration, VNS randomly selects an NS from the predefined list and applies it to generate a new solution. This new solution is then compared against the current solution. If the new solution shows improvement, it replaces the current solution, and the search resets to the first NS, continuing from the updated solution. Conversely, if no improvement is found, the algorithm proceeds to the next NS, continuing its search for a better solution until all NSs are evaluated. This iterative process allows VNS to effectively escape local optima and explore a wider search space (Almada-Lobo & James, 2010).

The original VNS algorithm, while versatile and effective for a wide range of optimization problems, can sometimes struggle with highly complex and specific challenges like those found in the VRP model. The weakness lies in the VNS's generalist approach to neighborhood selection, which might not always be efficient or targeted enough for problems with intricate constraints and objectives, leading to longer computational times and potentially suboptimal solutions. The SVNS algorithm addresses these weaknesses by introducing a tailored approach that significantly enhances performance for such specialized problems. SVNS uses two features: *Weighted NS selection policy* (used to prioritize the NSs at each iteration) and *directed NSs* (specifically designed for the complexities of food SCN design problem using the VRP model, enabling a more focused search that can navigate the problem space more effectively).

As shown in Figure 3, the SVNS algorithm begins with an initialization process, during which it first receives the input parameters, i.e., $nPop, Maxitr,\ P_{ns}$, where $Maxitr$ is the maximum number of iterations, $nPop$ is the population size, P_{ns} is the probability of performing NSs in the population at each iteration. Also, SVNS receives a set of neighborhood search structures $(\mathrm{ns} : 1,\ 2,\ \ldots,\ ns_{\max})$ including ordinary and directed structures used to find new solutions. Then, the number of times this algorithm tries the NSs is calculated as $NS_{\max} = [nPop \times P_{ns}]$. SVNS defines a selection weight for each NS based on their efficiency in improving the solutions in previous iterations, i.e., w_{ns}. Thus, the selection weights are calculated based on the number of times each NS has led

SVNS procedure()

Initialization
  Set parameters $(nPop, Maxitr, P_{ns})$;
  Receive a set of $N_{ns}(1, 2, \ldots, ns_{\max})$, including both ONSs and DNSs;
  Calculate No. of NSs as: $NS_{\max} = [nPop \times P_{ns}]$;
  $w_{nd} = \text{One}(1, ns_{\max})./ns_{\max}$;
  Generate the initial population (POP) with size $nPop$, and evalute them;
  Select the best solution as initial best solution S^*;
Main loop
  for $itr = 1$ to $Maxitr$ do
    for $pop = 1$ to $nPop$ do
      for $ns = 1$ to $NS_{\max}$ do
        $\pi_{\text{NS}} = \text{One}(1, ns_{\max})$; where π_{NS} shows No. improvement by each NS;
        S=POP(pop);
        Select a NS by RWS based on w_{ns};
        Perform the chosen NS on S, to find S';
        if S' is better than S^*, $S^* \leftarrow S'$, update π_{NS};
        elseif S' and S^* are equal, select one of them randomly update π_{NS};
        endif
      endfor and update POP(pop)
      $w_{ns} = \pi_{\text{NS}}./\text{sum}(\pi_{\text{NS}})$;
    endfor
    Sort (POP), $S^* = \text{POP}(1)$, and Reprts S^*
end

Figure 3. Pseudo code of SVNS.

to improving a solution, i.e., π_{NS}. The SVNS algorithm generates $nPop$ initial solutions.

In each iteration, a counter (π_{NS}) is generated with all elements initially set to one. If an NS improves the solution, its corresponding element in the counter is updated. At the end of each iteration, the counter matrix is normalized to produce the weight matrix (w_{ns}). Subsequently, the NSs are selected by RWS based on w_{ns}. The newly generated solutions are then compared with the current solutions, and the better one is selected for the next iteration. If both solutions are equal, one is chosen randomly to prevent the algorithm from getting stuck in a local optimum. At the end of each iteration, the current population is merged with the solutions obtained by NSs, and the $nPop$ best answers are kept, and the rest are discarded to keep the population constant.

NSs are classified as ordinary NSs (ONSs) and directed NSs (DNSs). The ONSs refer to general, problem-independent search mechanisms that can be applied across various optimization problems. These structures operate through basic local search operators to generate new solutions from existing ones. ONSs serve an exploratory role, enabling the algorithm to broaden the search space and reduce the risk of premature convergence. Since they do not incorporate problem-specific knowledge, ONSs provide a broad exploration mechanism that enables the algorithm to escape local optima and improve overall solution quality. This NS can be used for all sub-solutions. As shown in Figure 4, nine types of ONSs are used in SVNS as follows:

1. ***Inversion of two random numbers' positions (ITR):*** This operator reverses the sequence of elements between two randomly selected points in the solution. For example, in the vehicle routing sub-solution, this means reversing the order of retailer visits along a route. In ordering and discounting decisions, this approach can modify the sequence in which orders or discounts are applied to different products.
2. ***Adjacent pairwise exchange (APE):*** A randomly selected element is swapped with its adjacent element. This NS can be used for all sub-solutions. For example, in the vehicle routing sub-solution, this means exchanging the positions of two consecutive retailer visits, which can refine the travel route without drastically altering the structure.
3. ***Remove and place a number (R&P):*** A randomly chosen element is removed from its current position and placed at a new location within the sequence. In vehicle routing, this can shift a retailer visit from one position to another, balancing the vehicle load and optimizing delivery routes.
4. ***Shift neighborhood (SN):*** Two random numbers are selected, and if the first number is smaller, a forward shift is applied; otherwise, a backward shift is performed. This method helps fine-tune

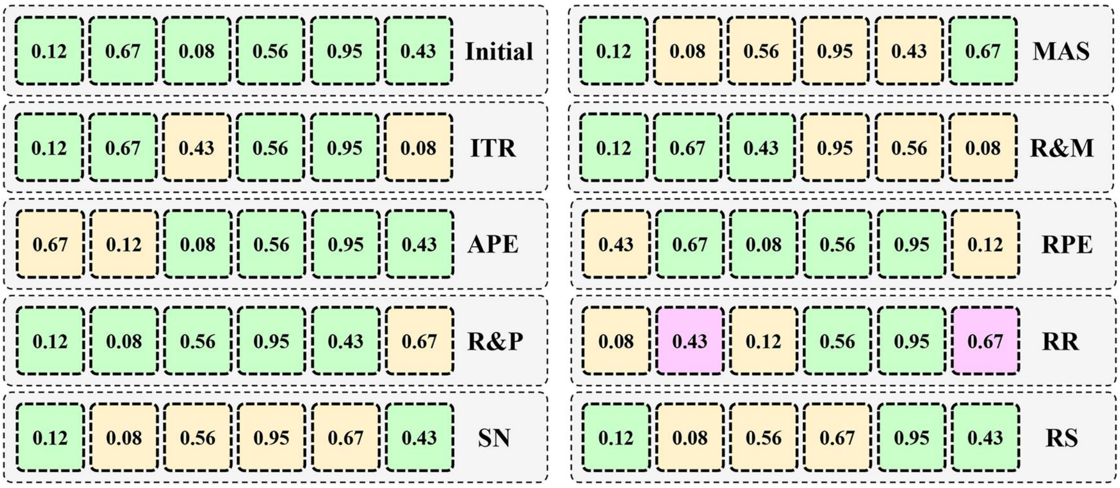

Figure 4. Ordinary neighborhood structures.

vehicle routes by adjusting retailer visit sequences and ensures better synchronization in pickup and delivery operations.

5. ***Moving a sequence of numbers (MAS)*:** A subsequence of elements is selected and moved to a new position within the solution. In VRP, this means relocating a group of retailers within a vehicle's route and improving delivery efficiency by clustering nearby locations.
6. ***Reversing and moving a sequence of numbers (R&M)*:** This structure operates like **MAS** but reverses the order of elements before relocating the sequence.
7. ***Random pairs of wise exchange (RPE)*:** Two randomly selected elements are exchanged. In the routing problem, this swaps the visit sequence of two retailers, which can improve vehicle utilization and reduce travel distance.
8. ***Random replacement (RR)*:** A set of randomly chosen elements undergoes multiple swaps, altering their positions. This approach is particularly effective in complex routing problems, where modifying multiple retailer visits at once can lead to significant efficiency gains.
9. ***Random shuffling in a sequence (RS)*:** A sequence of consecutive numbers is chosen, and then the positions of these numbers are randomly changed. Routing problems involve reordering a cluster of retailer visits to find a more optimal delivery sequence (Vahdani & Zandieh, 2010).

On the other hand, DNS is a problem-specific search mechanism designed to intelligently guide the search process toward promising regions of the solution space. Unlike ONSs, DNSs leverage domain-specific heuristics to make targeted modifications to a solution, improving its quality more efficiently. These structures focus on high-impact changes, such as balancing vehicle (BVL) loads, adjusting depot assignments, or fine-tuning

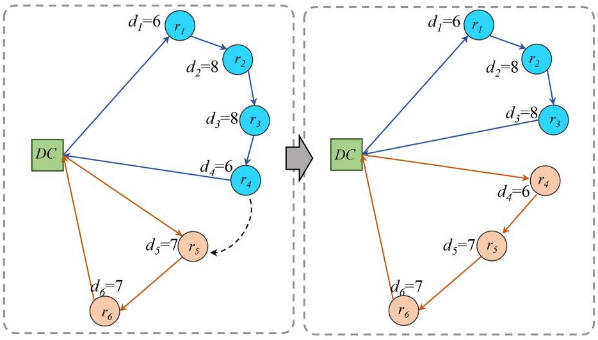

Figure 5. A schematic example for balancing vehicles' load (BVL).

inventory levels, ensuring that each move is strategically beneficial. By incorporating problem-specific knowledge, DNSs enhance the algorithm's ability to refine solutions faster, leading to improved convergence and better optimization results in complex vehicle routing and supply chain problems. The DNSs include seven types that are mentioned below:

4.2.1. Balancing vehicles' load (BVL)

The BVL neighborhood search specifically addresses the issue of unevenly distributed loads among vehicles, which can lead to inefficiencies such as longer delivery times and higher operational costs. Thus, as illustrated in Figure 5, an overloaded vehicle is selected, and then one of the retailers assigned to it is randomly selected. Then, the selected retailer is assigned to another vehicle with less load. BVL aims to achieve a more balanced distribution of loads across the fleet. Also, the randomness in retailer selection for reassignment introduces variability in solution space exploration, enhancing the algorithm's ability to escape local optima and find better overall solutions.

4.2.2. Adjusting manufacturers' load (AML)

The AML addresses the load imbalance among manufacturers. In this approach, the focus is on the

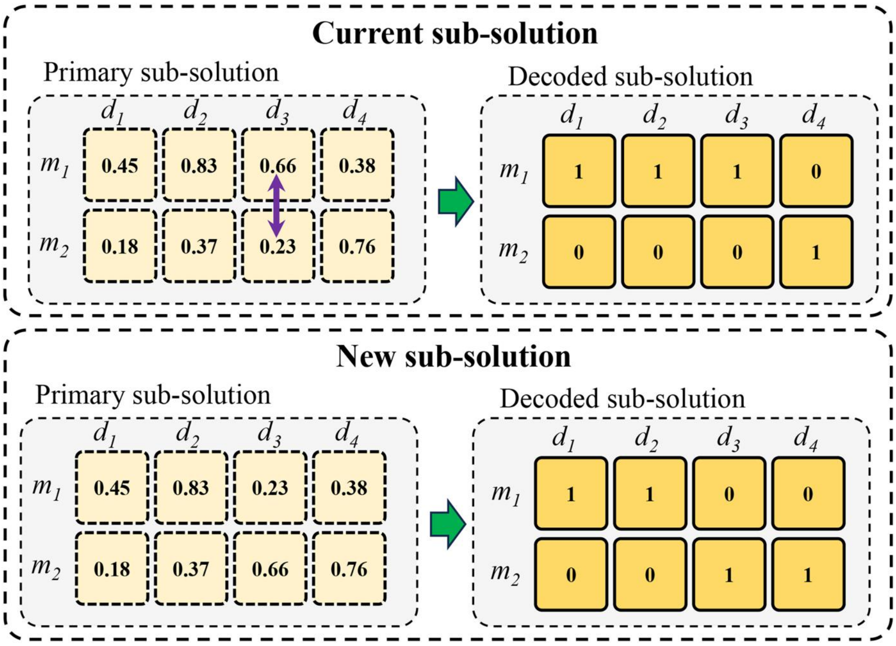

Figure 6. An example of adjusting manufacturers' load in assigning DCs to manufacturers' sub-solution.

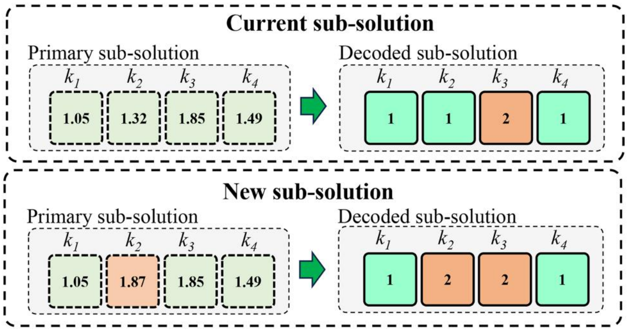

Figure 7. An example of a swap of depots (SOD) in assigning vehicles to DC sub-solution.

DCs and their allocation to manufacturers. Recognizing that some manufacturers may be overburdened by serving too many DCs, AML seeks to redistribute this load more evenly across the network. Thus, as shown in Figure 6, it identifies manufacturers at their maximum capacity regarding DC allocations and then randomly selects one of these overloaded centers to reassign it to another manufacturer associated with the fewest DCs.

4.2.3. Swap of depots (SOD)

In this policy, two depots are selected, and their allocated retailers are exchanged. The SOD neighborhood search selects two depots and then exchanges their allocated retailers. Thus, SOD seeks to change the distribution workload to find the best assignment according to their capacity. The underlying rationale, as shown in Figure 7, is that redistributing retailers between these extreme depots can lead to more efficient use of resources, such as vehicles and depot capacities, and ensure a more equitable distribution of delivery and pickup tasks.

4.2.4. Adjusting the load of depots/DC (ALD)

As illustrated in Figure 8, ALD neighborhood search identifies the DCs with the maximum and minimum volumes of product dispatched (sending), then selects a retailer assigned to the DC with maximum load, randomly. Then, the retailer is assigned to the DC with a minimum load. By doing so, ALD seeks to balance the distribution workload between depots to improve service levels across the network. The underlying rationale is that redistributing retailers between these extreme depots can lead to more efficient use of resources, such as vehicles and depot capacities, and ensure a more equitable distribution of delivery and pickup tasks.

4.2.5. Restrict the order of retailers (ROR)

The ROR neighborhood search tries to restrict the retailer's order with the most wasted products. Thus, as illustrated in Figure 9(a), if a retailer had the most unsold/deteriorated products in the previous period, the correction factor is increased to reduce this retailer's food waste by controlling the order amount.

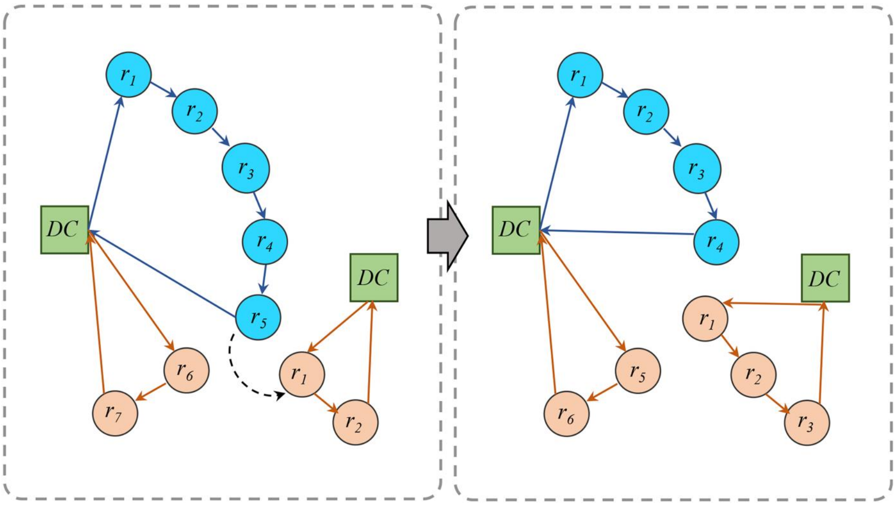

Figure 8. A schematic example for adjusting the depots/DC (ALD) load.

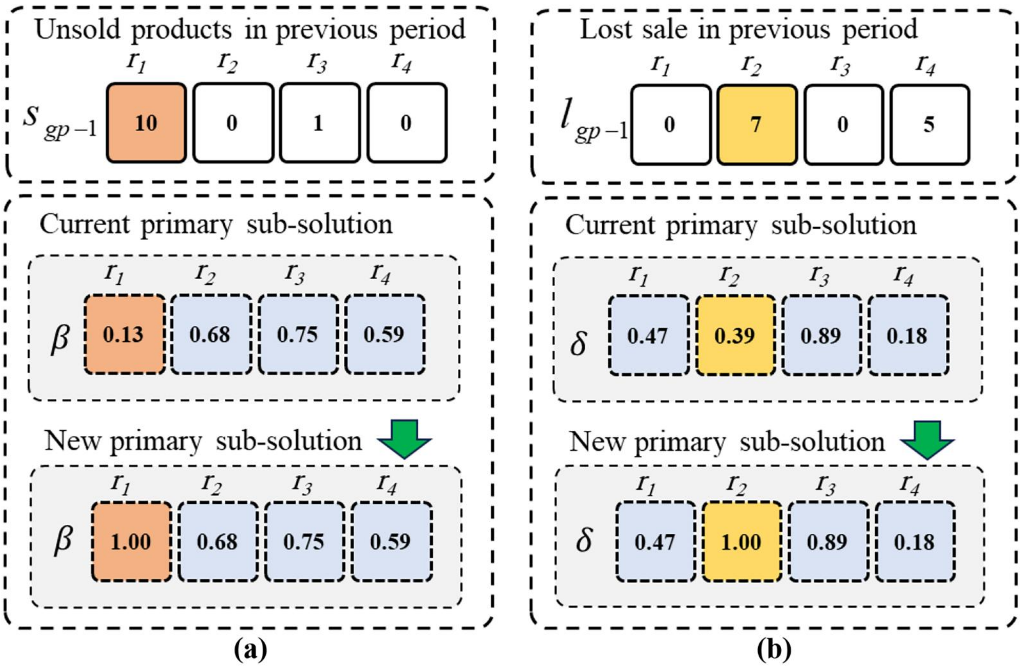

Figure 9. An example of restricting the order of retailers (ROR) and increasing the order of retailers (IOR).

Thus, the ROR operator adjusts future orders based on past unsold products to ensure that the retailer orders fewer products and reduces food waste.

4.2.6. Increasing retailers' orders (IOR)

This neighborhood search increases the retailer's orders due to the previous period's lost sales. Thus, as illustrated in Figure 9(b), if a retailer had the largest number of lost sales in the previous period, its correction coefficient δ_{rg} should be increased to reduce the lost sales. Thus, the IOR operator adjusts future orders based on past lost sales to ensure that the retailer orders more products to better align with customer demand and reduce the risk of future stockouts. The IOR helps minimize lost sales and optimize inventory levels by modifying order quantities based on past performance.

4.2.7. Adjusting retailers' discounts (ARD)

This NS adjusts retailers' discounting decisions based on their sales performance over the previous three periods. This approach aims to reduce both lost sales and unsold products by revising discount strategies according to past outcomes. Specifically, if a retailer experienced lost sales in previous periods, the ARD operator encourages offering discounts to attract more customers and increase sales. Conversely, if the retailer had unsold products, the ARD operator adjusts the strategy to remove or reduce discounts, as discounting in these cases may have failed to drive demand effectively.

5. Solution method validation

In this section, the validity of the proposed solution method is evaluated from two perspectives:

effectiveness and efficiency. Effectiveness refers to the ability to achieve the outputs correctly. The efficacy of an algorithm is often measured by its accuracy, precision, and the quality of the output it produces in practical applications. Also, efficiency relates to the algorithm's use of computational resources while achieving these results. This typically includes the computational time the algorithm uses to solve a problem. An efficient algorithm performs its tasks using minimal computational resources, thus making it scalable to larger datasets or more complex problem instances (Rabiee et al., 2012). Before that, we will explain how to generate data for the validations.

5.1. Data generation

To evaluate SVNS, a set of random problems is conducted involving analyzing various problems of different sizes, including manufacturers (M), DCs (D), retailers (R), and vehicles (K), see Table 1. The problems are classified into different sizes: small, medium, and large, based on the number of facilities. This classification directly reflects the complexity of the problem. Retailers were positioned randomly within a range from 0 to 100, and their distances were computed using the Euclidean norm. The demands of customers are generated by the normal distribution $\mathrm{N}(100, 20)$. Thus, six demand scenarios are defined for each problem, and accordingly, the demands are generated in six distances $\xi_1 = [-3\sigma, -2\sigma]$, $\xi_2 = [-2\sigma, -\sigma]$, $\xi_3 = [-\sigma, 0]$, $\xi_4 = [0, \sigma]$, $\xi_5 = [\sigma, 2\sigma]$, and $\xi_6 = [2\sigma, 3\sigma]$, with the possibility of $P(\xi_1) = 0.025$, $P(\xi_2) = 0.135$, $P(\xi_3) = 0.34$, $P(\xi_4) = 0.34$, $P(\xi_5) = 0.135$, and $P(\xi_6) = 0.025$. The number of periods is 100.

The data generated for the problem are shown in Table 2. Also, the capacity of the vehicle (Q) is determined as follows:

$$Q = (1+\eta)\frac{\sum_{p\in P}\sum_{r\in R}\sum_{g\in G} d_{rgp}(\xi)}{R \times P} \tag{32}$$

where η is a coefficient with $\mathrm{U}(0, 0.2)$. Also, the capacity of retailers for the Product g (Q_{rg}) is calculated as follows:

$$Q_{rg} = U\left[(1-\lambda)\frac{\sum_{p\in P}\sum_{r\in R} d_{rgp}(\xi)}{R \times P}, (1+\lambda)\frac{\sum_{p\in P}\sum_{r\in R} d_{rgp}(\xi)}{R \times P}\right] \quad \forall g \in G, r \in R \tag{33}$$

where λ is a coefficient with $\mathrm{U}(0, 0.2)$. Also, the capacity of manufacturers (Q_m) and DCs (Q_d) are generated in the same capacity as retailers.

Table 1. Size and level of problems.

Problem levels	Problem size ($M \times D \times R \times K$)	
Small scale	P.1.$(1 \times 1 \times 8 \times 2)$	P.5.$(1 \times 2 \times 24 \times 4)$
	P.2.$(1 \times 1 \times 10 \times 2)$	P.6.$(1 \times 3 \times 32 \times 4)$
	P.3.$(1 \times 1 \times 15 \times 3)$	P.7.$(1 \times 3 \times 36 \times 5)$
	P.4.$(1 \times 2 \times 20 \times 3)$	P.8.$(1 \times 4 \times 40 \times 5)$
Medium scale	P.9.$(2 \times 4 \times 42 \times 6)$	P.13.$(2 \times 6 \times 64 \times 7)$
	P.10.$(2 \times 5 \times 48 \times 6)$	P.14.$(2 \times 6 \times 68 \times 8)$
	P.11.$(2 \times 5 \times 52 \times 6)$	P.15.$(2 \times 6 \times 72 \times 8)$
	P.12.$(2 \times 5 \times 58 \times 7)$	P.16.$(2 \times 7 \times 78 \times 8)$
Large scale	P.17.$(3 \times 7 \times 80 \times 9)$	P.21.$(3 \times 9 \times 98 \times 11)$
	P.18.$(3 \times 8 \times 84 \times 9)$	P.22.$(3 \times 9 \times 102 \times 11)$
	P.19.$(3 \times 8 \times 88 \times 10)$	P.23.$(3 \times 10 \times 104 \times 12)$
	P.20.$(3 \times 8 \times 92 \times 10)$	P.24.$(3 \times 10 \times 110 \times 12)$

Table 2. Factors and their levels.

Factors	Levels	Factors	Levels
α	0.25	TC_{md}	$\sim dis_{ij} \times \varphi, \quad \varphi \sim \mathrm{U}(20, 40)$
C_{ij}	$\sim dis_{ij} \times \varphi, \quad \varphi \sim \mathrm{U}(1, 3)$	DC_{mg}	$\sim \mathrm{U}(3, 5)$
OC_d	$\sim \mathrm{U}(2, 5)$	ρ_g	$\sim \mathrm{U}(10, 18)$
RC_{dg}	$\sim \mathrm{U}(1, 3)$	θ_g	$\sim \mathrm{U}(2, 3)$
ϑ_{rg}	30%		

5.2. Parameter tuning

Most meta-heuristics define a set of parameters that need to be tuned. A good tuning of the values of that parameter can lead to the use of meta-heuristic capabilities to solve the desired problem (Montero et al., 2014). According to Zandieh and Gholami (2009), parameter tuning aims to find good values for the parameters before running the algorithm that remain constant during execution, and it is essential for enhancing the performance of meta-heuristic algorithms without producing performance variance under the influence of the external environment. Various approaches have been proposed to address this challenge, such as designing experiments (DOE), machine learning-based tuning, and metaheuristic self-tuning (Saremi et al., 2007; Huang et al., 2019; Talbi, 2021). DOE evaluates parameters at two levels (maximum and minimum), disregarding intermediate values, which limits its efficiency for continuous parameter tuning (Montgomery, 2017). Additionally, machine learning-based algorithms and metaheuristic self-tuning, which iteratively adjust parameters, are computationally intensive and time-consuming. Their performance is also highly dependent on the initial input parameters of the tuning module, whether based on meta-heuristic algorithms or machine learning techniques (Dobslaw, 2010; Yang et al., 2013). Thus, this article uses response surface methodology (RSM) to tune the parameters. The RSM method, introduced by Box and Wilson in 1951 as a statistical tool, is used to model and analyze the relationship between input parameters and output response (Nicolai et al., 2004). This method is widely used in industrial settings and parameter optimization to determine optimal parameter values (Rabiee et al., 2012). Some researchers have used the RSM method to set the parameters of meta-heuristic algorithms (Devika et al., 2014; Shadkam, 2022). This method has also shown good performance in VRP (Rabbani et al., 2018; Tohidifard et al., 2018; Govindan et al., 2019; Zhu & Hu, 2019).

The RSM method considers a lower bound (Xl) and an upper bound (Xh) for each variable. Thus, for an algorithm with k parameters, this method generates a set of experiments including three points:

- The factorial design points (n_f) are experiments that analyze factors at their high and low levels. The number of these points is equal to 2^k or a fraction of it. Main points.
- Axial Points (n_{ax}) provide valuable information about the curvature of models and analyze the axis of each design factor. The number of n_{ax} is equal to $2k$ coded ± 1 (Face entered).
- Central Points (n_{Cp}) check the repetition of the model and analyze factors at the middle level.

The number of central Points is known by n_{Cp} (Rabiee et al., 2012). The parameters of algorithms and their levels and number of experiments are presented in Table 3.

RSM simultaneously analyses the effectiveness and efficiency of algorithms by estimating appropriate parameters from the aspect of the objective function and computational time. The tuned values for parameters, R-squared (R2), and desirability are demonstrated in Table 4.

5.3. Algorithm comparison

5.3.1. Validation based on exact methods

A set of distinct small-scale test problems, including 16 random small problems, was designed with varying numbers of manufacturers (M), DCs (D), retailers (R), and vehicles (K). The performance of the objective function and the algorithm's efficiency were analyzed using the proposed solution approach. Each test instance was executed ten times in MATLAB 2022b on PCs equipped with an Intel Core i7 processor running at 3.00 GHz and 16 GB of RAM to evaluate the solutions obtained from SVNS. Additionally, a single execution was performed in GAMS to find the global solution. In total, 160 runs were conducted in MATLAB and 16 in GAMS. Table 5 provides a detailed overview of these test problems, including

Table 3. Level of factors in the algorithms with the number of experiments.

Algorithm	Factors and their levels (X_l, X_u)	$N_E = (n_f, n_{ax}, n_{cp})$
VNS	$MaxIt = (200, 600)$ $nPop = (150, 300)$ $P_{ns} = (0.6, 0.8)$	20=(2^3,6,6)
SA	$MaxIt = (200, 400)$ $nPop = (5, 10)$ $T_0 = (10, 20)$ $T_f = (0.001, 0.01)$	30=(2^4,8,6)
GA	$MaxIt = (200, 500)$ $Popsize = (100, 200)$ $P_c(0.65, 0.85)$ $P_m = (0.25, 0.45)$	30=(2^4,8,6)
SVNS	$MaxIt = (100, 300)$ $nPop = (100, 200)$ $P_{ons} = (0.05, 0.15)$ $P_{dns} = (0.05, 0.15)$	30=(2^4,8,6)

Popsize (population size), P_c (probability of crossover), P_m (probability of mutation), T_0 (initial temperature), T_f (final temperature)

Table 4. Tuned parameters, R-square (R^2), desirability (D).

Algorithm	Tuned parameters	R^2	D
VNS	$MaxIt = 576,\ nPop = 296, p_{ns} = 0.792$	%82.7	0.767
SA	$MaxIt = 357,\ Max_{ipt} = 7,\ n_{pop} = 2,\ T_0 = 10.45,\ T_f = 0.0011$	%69.2	0.681
GA	$MaxIt = 492,\ Popsize = 182, p_c = 0.742,\ p_m = 0.448$	%75.5	0.725
SVNS	$MaxIt = 296,\ nPop = 152\ P_{ons} = 0.096\ P_{dns} = 0.148$	%75.7	0.697

Table 5. Performance comparison of SVNS and GAMS across small-scale test problems.

Test ID	# M	# D	# R	# K	# P	# Runs by SVNS	GAMS run time (seconds)	SVNS Average run time (seconds)	GAMS	Minimum error (%)	Average error (%)	Maximum error (%)
1	1	1	10	2	10	10	12.0	38.2	5506	0.0	0.1	0.3
2	1	1	12	2	10	10	25.0	40.4	6430	0.0	0.2	0.5
3	1	1	14	2	20	10	83	38.6	9668	0.0	0.8	1.7
4	1	1	18	2	20	10	71	41.2	7379	0.0	1.0	1.9
5	1	1	18	3	20	10	830	41.4	11,058	0.9	1.6	2.5
6	1	1	20	3	20	10	701	43.1	11,753	0.5	1.1	2.0
7	1	1	24	3	20	10	1222	44.8	12,448	0.9	1.4	1.8
8	1	2	24	4	20	10	925	45.2	14,826	1.4	1.9	2.5
9	1	2	24	4	30	10	1143	43.8	16,166	1.2	1.8	2.4
10	1	2	28	4	30	10	1885	49.1	21,571	1.6	2.2	3.0
11	1	2	32	4	30	10	1231	51.9	23,454	1.3	2.2	2.9
12	1	2	36	4	30	10	1376	52.8	26,605	1.1	1.8	2.7
13	1	2	40	4	30	10	1818	51.3	25,730	1.3	2.1	3.0
14	1	2	42	5	30	10	1621	53.0	28,918	1.7	2.7	3.3
15	1	3	42	5	40	10	2155	53.9	33,890	1.2	2.1	3.1
16	1	3	48	5	40	10	–	54.9	–	–	–	–

the number of manufacturers, DCs, retailers, vehicles, periods, number of runs by SVNS algorithm, GAMS runtime (in seconds), MATLAB runtime (in seconds), objective function values obtained by GAMS, and the minimum, average, and maximum percentage errors.

From a performance standpoint, SVNS demonstrates superior efficiency, particularly in runtime, when addressing complex problems. This is evident in the resolution of the most challenging test case, i.e., the sixteenth problem, where AIWO completed the task in around 55 s. In contrast, the GAMS solver failed to find a solution after forty minutes, highlighting SVNS's significant efficiency advantage. Additionally, regarding effectiveness, SVNS consistently delivers accurate results, with deviations from the expected outcomes measured as minimum, average, and maximum percentages. These deviations are reported at less than 1.7%, 2.7%, and 3.3%, respectively. Furthermore, AIWO achieved optimal results for the smallest test cases with a difference close to zero.

5.3.2. Benchmark algorithms

The SVNS algorithm is compared with three benchmark algorithms, namely, GA and simulated annealing (SA), as two well-known algorithms for solving VRP problems (Ariyani et al., 2018), and the original VNS as the base algorithm to evaluate its performance. To ensure a fair comparison, GA, SA, and VNS algorithms are optimized and designed to address the specific characteristics of the NTE FSC as follows:

Benchmark 1: GA: GA is a widely used metaheuristic inspired by natural selection and evolution principles. It operates through key processes such as selection, crossover, and mutation to explore and exploit the solution space, making it a powerful tool for solving complex combinatorial optimization problems (Guner Goren et al., 2010). As GA is a well-established method in the literature and widely recognized for solving routing, inventory management, and facility location decisions within SCN design problems (Karaoglan & Altiparmak, 2010; Agrawal et al., 2022), it has been selected as a benchmark algorithm for this problem. Additionally, for a fair comparison with the proposed SVNS algorithm, we have aligned the NSs in SVNS with the mutation operations in GA. This alignment ensures that both algorithms explore the solution space using a comparable search mechanism.

Benchmark 2: SA: SA is a well-established metaheuristic optimization technique inspired by the annealing process in metallurgy, where materials are heated and then slowly cooled to reach a minimum energy state. The algorithm simulates this process through a cooling schedule, which helps it avoid premature convergence to local optima by allowing occasional uphill moves during the search process (Golden & Skiscim, 1986). SA has been selected as a benchmark algorithm due to its widespread application and proven effectiveness in solving VRP and SCN design problems (Ross & Jayaraman, 2008). To ensure a fair comparison between SA and SVNS, we adopted the same NSs in SVNS as those used in SA's neighborhood search process. This alignment provides a consistent basis for evaluating the performance of both algorithms in exploring and exploiting the solution space.

Benchmark 3: Original VNS: We consider the original VNS a benchmark that uses only ordinary NSs as general operators throughout the search process. Unlike SVNS, which incorporates general and problem-specific NSs, the original VNS relies solely on general NSs. This baseline configuration allows us to isolate and evaluate the impact of problem-specific NSs introduced in SVNS. The comparison highlights how including problem-specific, directed NSs in SVNS enhances search efficiency and solution quality.

5.3.3. Comparative performance analysis of algorithms

This section systematically evaluates the performance of algorithms using the proposed criteria as benchmarks for comparative analysis. The main objective is to determine whether there are significant differences in the performance of different algorithms. The obtained data (objective function and CPU time) are converted into RPD values to evaluate the proposed algorithms. The average RPD results of the expected objective function and CPU time are calculated in Table 6. The low p values (0.000) and high F-values (35.13 for the objective function and 106.99 for computational time) indicate a statistically significant difference between the algorithms in terms of both solution quality and computational efficiency. The RPD value is calculated as follows:

$$RPD = \frac{|\mathrm{Alg}_{obj} - \mathrm{Min}_{obj}|}{\mathrm{Min}_{obj}} \tag{34}$$

where Alg_{obj} is the obtained solution for each algorithm, and Min_{obj} is the best result of the algorithm. Also, the same formula is used to calculate the RPD of computational time, where the CPU time is used instead of the objective function. A lower RPD value indicates that the observed performance is closer to the expected performance. In this regard, Fisher's least significant difference (LSD) method was used

Table 6. ANOVA results for the objective function and computational time.

	Source	DF	SS	MS	F	p Value
Objective function	Factor	3	0.18425	0.06142	35.13	0.000
	Error	92	0.16084	0.00175		
	Total	95	0.34509			
Computational time	Factor	3	6.8175	2.2725	106.99	0.000
	Error	92	1.9542	0.0212		
	Total	95	8.7717			

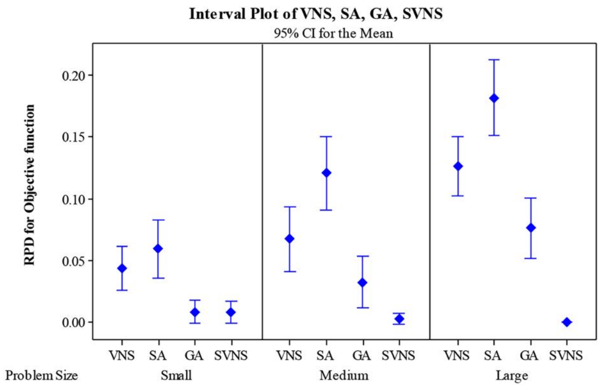

Figure 10. Plot the mean and LSD intervals for the algorithms GA, SA, VNS and SVNS.

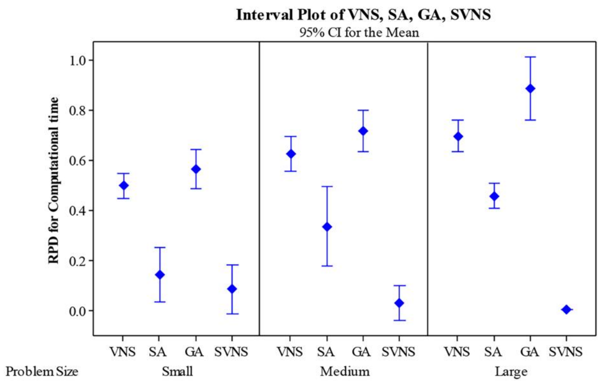

Figure 11. Plot the mean and LSD intervals for GA, SA, VNS, and SVNS algorithms.

to compare the proposed algorithms in pairs. The LSD method is a statistical procedure used to compare multiple groups to determine if there are significant differences between them. Thus, LSD has been applied to compare all algorithms regarding their objective function and computational times. The LSD method results presented in Table 6 reveal a significant difference between the algorithms in terms of both the objective function and computational time. Each pairwise comparison shows a significant difference (Yes), as none of the confidence intervals include zero, indicating that the algorithms exhibit distinct performance levels in solution quality and computational efficiency.

In addition, a statistical assessment was conducted with a 95% confidence interval for the objective function and computational time to evaluate the significance of the results. According to Figure 10, SVNS outperformed the other algorithms from an objective function perspective. Also, according to Figure 11, SVNS has a better computational time than the benchmarks. These figures show that the upper and lower limits of the SVNS algorithm do not overlap with the benchmarks in the medium and large sizes. Moreover, Table 7 shows that SVNS is statistically superior to other algorithms, i.e., GA, SA, and VNS, based on the objective function and computational time. In addition, in Figure 12, the improvement of algorithm responses can be seen. This figure shows improved vehicle routes from each DC (squares) to retailers (circles).

6. Case study

Carrefour, a prominent French multinational retail and wholesaling corporation, is the focal point for this case study to demonstrate the practical application of the proposed model. Carrefour is one of France's top four food retail groups and ranks eighth globally in revenue. The company operates across 10 countries directly and collaborates with retailers in another 20 countries, collectively working with over 5000 retailers worldwide. This broad operational reach positions Carrefour as a significant global FSC player. The case study focuses explicitly on Carrefour's operations related to NTE food products within Rennes, a city in the Ille-et-Vilaine department of the Brittany region in northwestern France. Rennes had a population of 222,485 as of the 2018 census, providing a relevant context for examining urban FSC dynamics. Carrefour's NTE food distribution activities in this region align with broader food waste reduction efforts implemented across the company.

According to Carrefour (2022) reports, the company employs several strategies to minimize food waste while maintaining product accessibility. These initiatives include offering substantial discounts on NTE products, typically 30–60%. For example, items nearing expiration dates are marked down to incentivize purchases and reduce potential waste. Additionally, Carrefour has introduced innovative solutions like "Zero Gaspi" (zero waste) or anti-waste packaging, applied to slightly wilted vegetables or other items that remain safe and consumable despite minor imperfections. Products in these packages are generally sold with a 50% discount. These

Table 7. Fisher 95% individual confidence intervals all pairwise comparisons for problems.

	Algorithms	Lower	Upper	Significant difference at 95% level
Objective function	VNS and SA	0.01771	0.06566	Yes
	VNS and GA	−0.0641	−0.01616	Yes
	VNS and SVNS	−0.09944	−0.05149	Yes
	SA and GA	−0.10579	−0.05784	Yes
	SA and SVNS	−0.14112	−0.09317	Yes
	GA and SVNS	−0.05931	−0.01136	Yes
Computational time	VNS and SA	−0.3786	−0.2115	Yes
	VNS and GA	0.0329	0.2000	Yes
	VNS and SVNS	−0.6518	−0.4847	Yes
	SA and GA	0.3279	0.4950	Yes
	SA and SVNS	−0.3568	−0.1896	Yes
	GA and SVNS	−0.7682	−0.6011	Yes

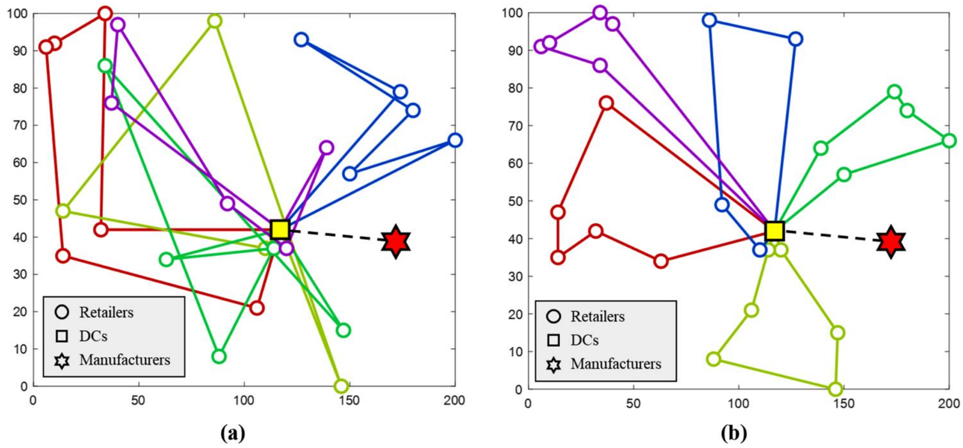

Figure 12. Improvement of problem P.5. ($1 \times 1 \times 5 \times 25$) by SVNS in (a) 20th and (b) 200th iterations.

discounting strategies reduce food waste and cater to cost-conscious consumers, creating a win-win situation for the company and its customers. The insights from this case study provide a real-world foundation for validating the proposed model's effectiveness in minimizing waste while maintaining operational efficiency in FSCs.

The corporation is active in organic food by manufacturing various products under the Carrefour brand. Carrefour has implemented diverse store formats to accommodate its customers' varied needs and preferences. This study considers four types of stores:

- *Hypermarkets* are large-scale retail stores offering various products, including groceries, household items, electronics, clothing, etc.
- *Carrefour Markets* are smaller than hypermarkets and are designed to serve urban or suburban areas. Although they offer fewer products than hypermarkets and markets, they still provide a comprehensive range of groceries, fresh produce, and household essentials.
- *Carrefour Express* is a convenience-focused store in urban centers, train stations, or other areas with high foot traffic. It prioritizes convenience by offering a limited selection of products, focusing on grab-and-go items, snacks, beverages, and ready-to-eat meals.
- *Carrefour City:* similar to express stores, they are designed for urban areas where space is at a premium. They stock fresh foods, snacks, beverages, and other essentials.

The distribution of Carrefour stores within Rennes and their type are illustrated in Figure 13. According to the figure, 26 Carrefour stores are considered in this case study, including one Hypermarket, four Carrefour Markets, seven Carrefour Express, and fourteen Carrefour City.

In the case of the problem, Carrefour's distribution network includes a manufacturer, two DCs, and 26 retailers to ensure efficient and reliable distribution of these products throughout the city. In this case problem, the Le Rheu Carrefour platform, located 9 km southwest of Rennes, is introduced as the manufacturer that produces food products and procures goods from local farmers and ranchers. Product pallets are taken out of Le Rheu early in the morning and sent to DCs to be prepared and delivered to retailers in the city. Food products are sent to retailers through two DCs, the north DC (DC Les Gayeulles) and the south DC (DC Av. du Canada). These DCs include the docks for loading and unloading trucks and parking for trucks.

The trucks in this supply chain are specifically designed to handle the distribution of fresh foods,

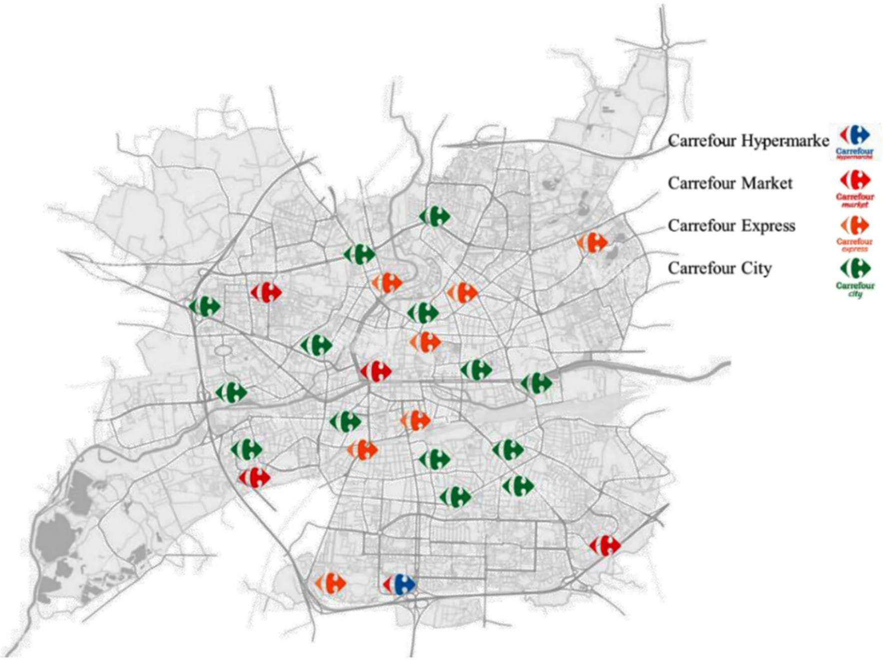

Figure 13. Dispersion of Carrefour retailers in the city of Rennes.

including fruits, vegetables, dairy, and meats, by maintaining controlled temperatures to preserve product freshness and safety. These vehicles have advanced refrigeration systems that ensure optimal storage conditions throughout transport. Two trucks of different sizes are utilized in this network. Large trucks are used to transport products from the Le Rheu Carrefour platform to the two DCs by direct shipment strategies. Also, medium-sized trucks handle deliveries from the DCs to retailers across the city. Their medium-sized truck allows easy navigation through narrow city streets, enabling efficient deliveries to densely populated urban areas. The company operates a limited fleet of medium-sized refrigerated vehicles between DCs and retailers designated for distributing shipments within urban areas. The distances between nodes are obtained from an external database, specifically Google Maps. Based on available data, the average transportation cost for large trucks, which use direct shipments from the Le Rheu Carrefour platform to the DCs (TC_{md}) carried out by transportation companies, is higher, estimated at 30 euros per unit of distance. Also, the average transportation cost for medium-sized trucks used for urban distribution from DCs to retailers (C_{ij}), which is carried out by the company's trucks, is estimated at 2 euros per unit of distance.

Furthermore, standard packaging for fruits and vegetables typically uses durable, stackable boxes with dimensions of $22 \times 15 \times 5.5$ inches, designed to maximize space efficiency during transport while protecting perishable products. These boxes are made from corrugated cardboard or plastic, ensuring proper ventilation, structural integrity, and compatibility with standard pallet configurations. The large trucks have a capacity of 600 standard boxes, and the medium-sized trucks have a capacity of 80 standard boxes. Three NTE products examined in the case problem include washed lettuce, ready salad, and vegetables packaged with the Carrefour brand. The requests for washed lettuce and vegetables are the same, and the request for ready salad is a quarter of them. The Carrefour stores, their type, and the demand scenarios as standard boxes for washed lettuce are presented in Table 8. A standard box can hold ten washed lettuces, ten vegetable items, or eight salad packages. We consider 100 periods as working days. The discount level for all products in all retailers is considered 40%. Also, it should be mentioned that the euro is considered the monetary unit. Other input parameters are presented in Table 9.

7. Results and discussion

The best network topology for the case problem in period one is presented in Figure 14. The lines in the figure only show the nodes' connections and do not represent the real path. According to this figure, the retailers in the northern part of the city were connected to the North DC (DC Les Gayeulles), and the retailers in the southern part of the city were connected to the South DC (DC Av. du Canada). Thus, placing DCs in positions that ensure they are roughly equidistant from various retailers is advisable. This setup minimizes transportation time and costs and can significantly improve logistics efficiency. One effective approach might be to position a single DC in the city's center if the urban layout

Table 8. Carrefour stores, their type, and their demand scenarios.

r	Carrefour	Type	Demand Scenario 1 (SD1) 5%	Demand Scenario 2 (SD2) 15%	Demand Scenario 3 (SD3) 30%	Demand Scenario 4 (SD4) 30%	Demand Scenario 5 (SD5) 15%	Demand Scenario 6 (SD6) 5%
1	Av. du Canada	Hypermarket	U(24,31)	U(31,39)	U(39,46)	U(46,53)	U(53,61)	U(61,68)
2	Rue de Brest	Market	U(16,19)	U(19,21)	U(21,24)	U(24,27)	U(27,29)	U(29,32)
3	Centre commercial La Poterie	Market	U(18,21)	U(21,24)	U(24,27)	U(27,30)	U(30,33)	U(33,36)
4	President John F.Kennedy	Market	U(15,18)	U(18,20)	U(20,23)	U(23,25)	U(25,28)	U(28,30)
5	Rue d'Isly	Market	U(16,19)	U(19,21)	U(21,24)	U(24,27)	U(27,29)	U(29,32)
6	Rue Saint-Malo	Express	U(6,7)	U(7,8)	U(8,9)	U(9,10)	U(10,11)	U(11,12)
7	Rue d'Antrain	Express	U(5,7)	U(7,8)	U(8,10)	U(10,12)	U(12,13)	U(13,15)
8	Pl. Albert Bayet	Express	U(6,7)	U(7,7)	U(7,8)	U(8,9)	U(9,9)	U(9,10)
9	Pl. de Bretagne	Express	U(6,7)	U(7,7)	U(7,8)	U(8,9)	U(9,9)	U(9,10)
10	Rue de Nantes	Express	U(5,6)	U(6,6)	U(6,7)	U(7,7)	U(7,8)	U(8,8)
11	Rue Maréchal Joffre	Express	U(5,6)	U(6,7)	U(7,8)	U(8,8)	U(8,9)	U(9,10)
12	route De Fougeres	Express	U(6,7)	U(7,7)	U(7,8)	U(8,9)	U(9,9)	U(9,10)
13	Av. des Monts d'Arrée	City	U(4,4)	U(4,5)	U(5,5)	U(5,5)	U(5,6)	U(6,6)
14	Bd de Verdun	City	U(3,4)	U(4,5)	U(5,6)	U(6,6)	U(6,7)	U(7,8)
15	Rue Saint-Malo	City	U(5,6)	U(6,7)	U(7,8)	U(8,8)	U(8,9)	U(9,10)
16	Mail Louise Bourgeois	City	U(4,5)	U(5,5)	U(5,6)	U(6,7)	U(7,7)	U(7,8)
17	Rue de Fougères	City	U(4,5)	U(5,6)	U(6,7)	U(7,7)	U(7,8)	U(8,9)
18	Av. Sir Winston Churchill	City	U(5,5)	U(5,5)	U(5,6)	U(6,6)	U(6,6)	U(6,6)
19	Av. Aristide Briand	City	U(6,6)	U(6,7)	U(7,7)	U(7,7)	U(7,8)	U(8,8)
20	Rue de Lorient	City	U(5,6)	U(6,6)	U(6,7)	U(7,7)	U(7,8)	U(8,8)
21	Rue de l'Alma	City	U(5,6)	U(6,6)	U(6,7)	U(7,7)	U(7,8)	U(8,8)
22	Bd de Metz	City	U(5,6)	U(6,6)	U(6,7)	U(7,7)	U(7,8)	U(8,8)
23	Bd Voltaire	City	U(3,4)	U(4,5)	U(5,7)	U(7,8)	U(8,9)	U(9,10)
24	Av. Sergent Maginot	City	U(4,5)	U(5,6)	U(6,7)	U(7,7)	U(7,8)	U(8,9)
25	Bd Léon Bourgeois	City	U(4,5)	U(5,6)	U(6,7)	U(7,8)	U(8,9)	U(9,10)
26	Bd Georges Clemenceau	City	U(4,5)	U(5,6)	U(6,7)	U(7,8)	U(8,9)	U(9,10)

Table 9. Factors and their levels.

Title	Type	Input data
Manufacturer (M)	Le Rhu	$Q_1 = 800$
		$DC_{11}, DC_{12}, DC_{13} = 1$
Distribution centers (DC)	North DC (DC Les Gayeulles)	$Q_1 = 385$
		$OC_1 = 2$
		$RC_{11} = 0.25, RC_{12} = 0.25, RC_{13} = 0.25$
	South DC (DC Av. du Canada)	$Q_2 = 415$
		$OC_2 = 2$
		$RC_{21} = 0.25, \quad RC_{12} = 0.25, \quad RC_{23} = 0.25$
Retailer (R)	Hypermarket	$Q_{r1} = 60, \quad Q_{r2} = 15, \quad Q_{r3} = 60$
	Market	$Q_{r1} = 30, \quad Q_{r2} = 8, \quad Q_{r3} = 30$
	Express	$Q_{r1} = 10, \quad Q_{r2} = 3, \quad Q_{r3} = 10$
	City	$Q_{r1} = 8, \quad Q_{r2} = 2, \quad Q_{r3} = 8$
Products (G)	Washed lettuce packages	$\rho_1 = 1, \ \theta_1 = 1$
	Ready salad packages	$\rho_2 = 3, \ \theta_2 = 1.5$
	Vegetable packages	$\rho_3 = 1, \ \theta_3 = 1$

allows easy access from all directions. Alternatively, placing several DCs around the city can distribute products more evenly, ensuring quicker delivery times and better service to a wider area.

As this study emphasizes the importance of retailers in reducing total costs and food waste, the retailer's decisions are examined in this section. Retailers can influence the supply chain through two policies: smoothing ordering and discounting. In policy smoothing ordering, there are two decision variables, which are β_{rg} as the constant of wasted products and δ_{rg} as the constant of lost sales. Figure 15(a) shows the interaction effects of β_{rg} and δ_{rg} on the total cost. According to this figure, increasing the constants associated with wasted products and lost sales negatively impacts the total cost by influencing future order quantities in the smoothing ordering policy. When these constants are increased, the model responds more aggressively to past waste and lost sales by adjusting future orders. Specifically, higher constants for wasted products reduce future order quantities to minimize excess inventory, potentially leading to understocking and lost sales. Conversely, increasing the constants for lost sales leads to larger future orders to meet potential demand, potentially resulting in overstocking and increased waste. This dynamic adjustment creates inefficiencies, such as higher transportation, holding, and disposal costs, ultimately driving the total cost.

Also, we analyze the interactive effect of the discount level (ϑ_{rg}) in retailers related to discounting policy and β_{rg} as the constant of wasted products. As shown in Figure 15(b), increasing discount levels and the constant of wasted products lead to a clear negative impact on the total cost due to the dynamics of inventory management and consumer

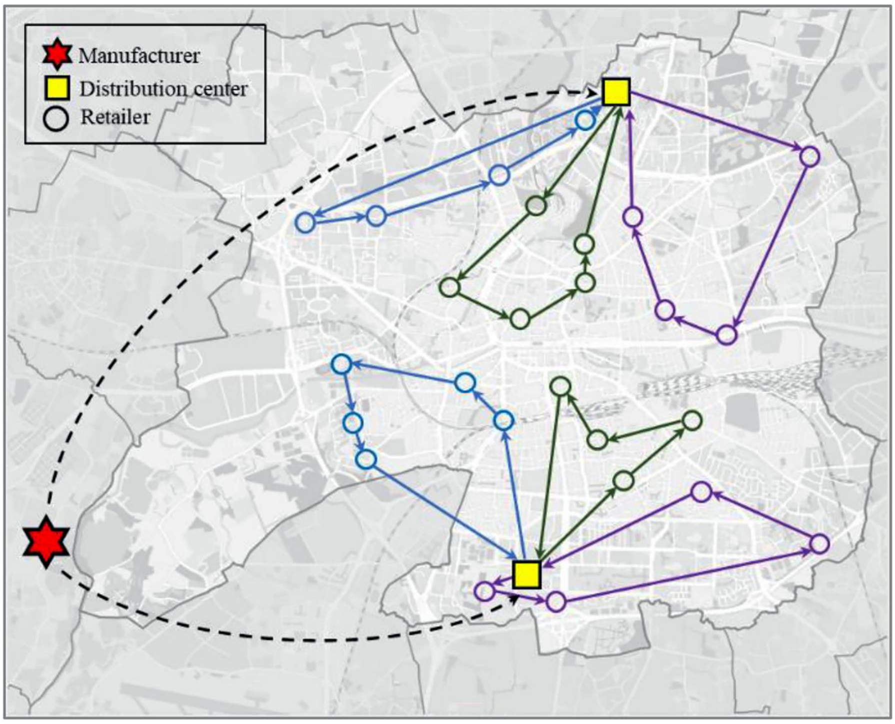

Figure 14. A schematic of urban transportation in the case problem (in period 1).

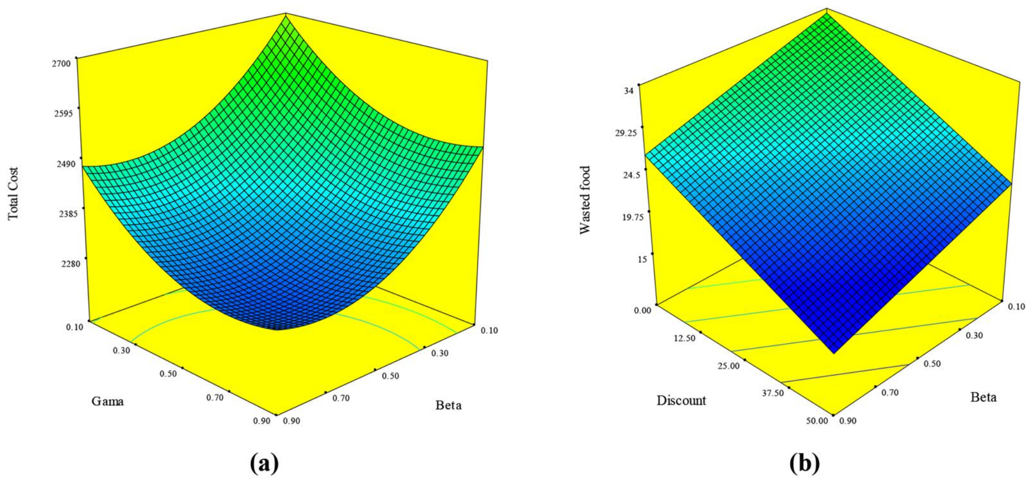

Figure 15. The 3D plot of the interaction of (a) β_{rg} and δ_{rg} for total cost and (b) ϑ_{rg} and β_{rg} for food waste.

behavior. When discount levels rise, retailers offer more price reductions to stimulate demand for NTE products, which reduces waste but simultaneously lowers revenue, increasing overall costs if discounts are applied excessively. Additionally, increasing the constant for wasted products prompts the system to adjust future orders based on historical waste, often resulting in smaller, more conservative orders. This strategy reduces unsold inventory, leading to lower ordering and transportation costs.

Also, as the supply chains focus more on reducing total costs, reducing total costs is chosen as the objective function in this article. However, achieving food waste reduction and reducing total costs requires multi-objective decision-making (MODM) approaches. Thus, the utility function introduced by Derringer and Suich (1980) is used to optimize multiple objectives as follows:

$$d(obj_i) = \left(\frac{UOBJ_i - OBJ_i}{UOBJ_i - LOBJ_i}\right)^S \quad LOBJ_i \le OBJ_i \le UOBJ_i \tag{35}$$

where OBJ_i is objective, and obj_i is the utility function of OBJ_i in the form of minimization, $LOBJ_i$ and $UOBJ_i$ are lower and upper bounds of OBJ_i obtained for different levels of input parameters, respectively. In this section, S is the weight of OBJ_i known as the severity of that. Thus, the desirability is calculated as:

$$D = \sqrt{d(obj_1) \times d(obj_2) \times \cdots \times d(obj_\omega)} \tag{36}$$

where ω is the number of objectives. In this research, the severity of total cost and wasted food products are 3 and 1, respectively. Accordingly, a sensitivity analysis was performed on two variables β_{rg} and δ_{rg} in the case problem, the result is shown in Figure 16. If both are selected at their high level, it leads to an increase in desirability, which indicates a reduction in total costs and food waste.

Moreover, a single-factor sensitivity analysis was performed on the discount rate. If all parameters are fixed and the discount rate changes from 0% to 50%, as shown in Figure 17(a), the total cost decreases and then increases. This means that the best discount level to reduce the total costs is about 30% of the product price from the total cost perspective. As anticipated and illustrated in Figure 17(b), the amount of wasted goods decreases consistently as the discount rate increases. We apply a MODM approach to determine the optimal discount level for total cost and wasted products perspective. As shown in Figure 17(c), when analyzing the impact of the discount rate on desirability, the optimal discount level that simultaneously minimizes total cost and wasted food products is approximately 40% discount on the product price. According to the results, the practitioners should consider the broader implications of waste reduction, such as environmental impact, brand reputation, and compliance with food waste regulations, rather than focusing solely on cost. Implementing a dynamic discounting strategy that adjusts discount levels based on inventory levels, demand patterns, and shelf life can optimize cost efficiency and waste reduction.

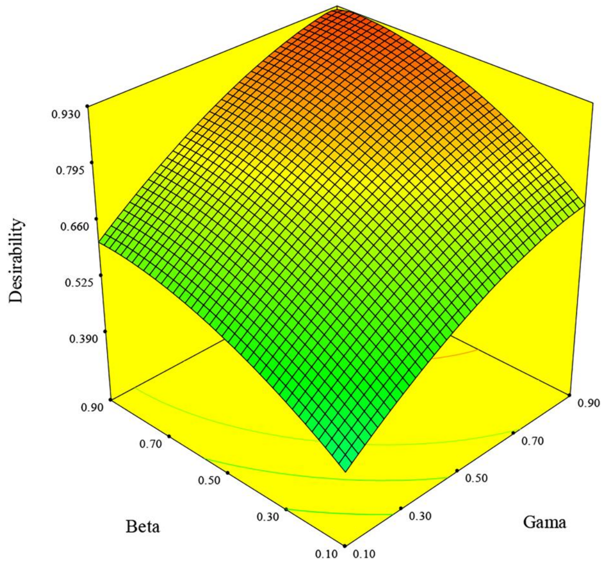

Figure 16. The 3D plot of the interaction of β_{rg} and δ_{rg} for desirability.

8. Conclusion and future research

This study introduces a novel optimization technique for designing FSC networks that incorporate near-expiration products. The proposed model considers the influence of retailers on cost reduction and food waste minimization by incorporating a smoothing ordering policy and discounting. In this model, retailers respond to customer demand and employ smoothing ordering and discounting policies to optimize order levels, thereby diminishing food waste and overall costs. Products are transported from manufacturers to DCs *via* direct shipment and routed to retailers using vehicle routing strategies to minimize total network costs. The formulation of this FSC network involves a two-echelon transportation problem, which integrates ordering and discounting policies within the retail sector. Given the complexity of this issue, we have developed a new metaheuristic algorithm, SVNS. This algorithm leverages directed and ordinary NSs, selected based on a prioritization approach that assesses their effectiveness in prior iterations. Several experimental scenarios are constructed to evaluate the performance of the proposed algorithm, and the results are benchmarked against established methods to demonstrate the efficacy and efficiency of our approach.

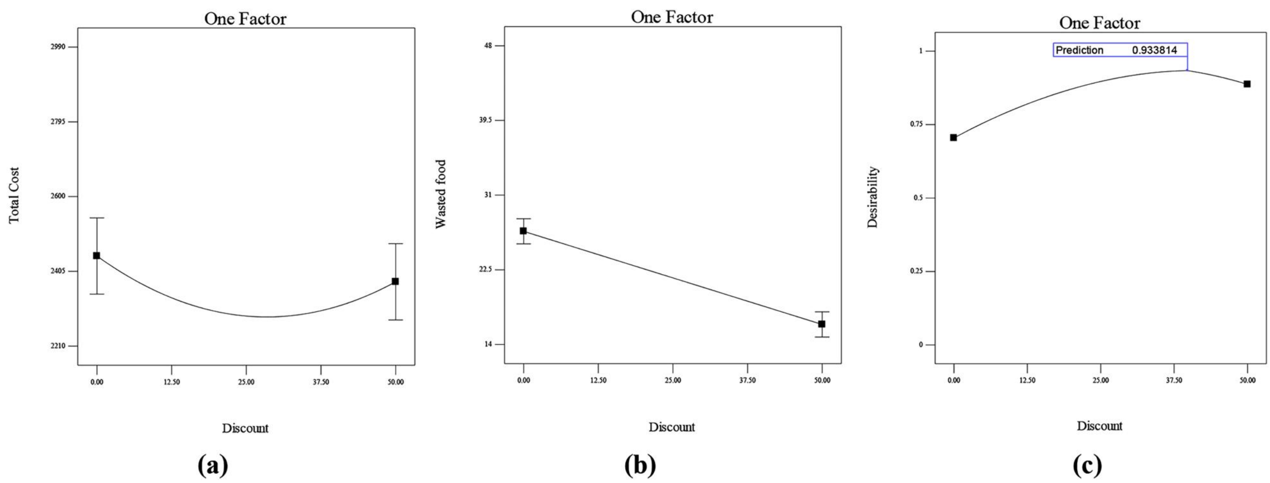

Figure 17. The plot of the discount level for total cost and desirability.

According to the results, SVNS outperforms the benchmark methods in total cost performance measures. The optimal topology for the distribution network includes situating DCs that are equidistant from retailers. Also, analyzing retailer policies shows that smoothing ordering and discounting significantly impact cost and waste. The constant of wasted products as a correction factor directly relates to food waste and the total cost and should be selected at a high level. Also, for this problem, the 30% discount rate can be considered an optimal level of discounting that can minimize total costs, and the 40% discount rate can maximize desirability, minimizing both total costs and wasted food products. A direct direction for further research is to consider the option of contracts allowing retailers to return the products at the end of the day. Although product prices are increased in these contracts, food waste can be reduced. Also, a decision support system can be developed for retailers to help them systematically forecast the demand, order to DCs, and decide about discounting.

Notes

1. https://www.tescoplc.com/.
2. https://www.paragonrouting.com/.

Disclosure statement

Authors declare that they have no conflict of interest.

Funding

No funding was received.

ORCID

Thi Le Hoa Vo http://orcid.org/0000-0001-6737-4267
Madjid Tavana http://orcid.org/0000-0003-2017-1723
Emmanuelle Fromont http://orcid.org/0000-0002-9847-8576

Data availability statement

Data will be made available upon request.

References

Adenso-Díaz, B., Lozano, S., & Palacio, A. (2017). Effects of dynamic pricing of perishable products on revenue and waste. *Applied Mathematical Modelling*, *45*, 148–164. https://doi.org/10.1016/j.apm.2016.12.024

Agrawal, A. K., Yadav, S., Gupta, A. A., & Pandey, S. (2022). A genetic algorithm model for optimizing vehicle routing problems with perishable products under time-window and quality requirements. *Decision Analytics Journal*, *5*, 100139. https://doi.org/10.1016/j.dajour.2022.100139

Alavi, S., Azad, N., Heydar, M., & Davoudpour, H. (2016). Integrated production, inventory, and location-allocation decisions in designing supply chain networks. *International Journal of Information Systems and Supply Chain Management*, *9*(4), 22–42. https://doi.org/10.4018/IJISSCM.2016100102

Almada-Lobo, B., & James, R. J. (2010). Neighbourhood search meta-heuristics for capacitated lot-sizing with sequence-dependent setups. *International Journal of Production Research*, *48*(3), 861–878. https://doi.org/10.1080/00207540802446787

Alrashidi, M., & Al Ghamdi, M. (2024). *Variable neighborhood search based on reinforcement learning for green vehicle routing problem.* In *International Conference on Artificial Intelligence in Information and Communication* (ICAIIC, 19-22, February 2024), Osaka, Japon.

Ariyani, A. K., Mahmudy, W. F., & Anggodo, Y. P. (2018). Hybrid genetic algorithms and simulated annealing for multi-trip vehicle routing problem with time windows. *International Journal of Electrical & Computer Engineering*, *8*(6), 2088–8708. http://doi.org/10.11591/ijece.v8i6.pp4713-4723

Bezerra, S. N., de Souza, S. R., & Souza, M. J. F. (2023). A general VNS for the multi-depot open vehicle routing problem with time windows. *Optimization Letters*, *17*(9), 2033–2063. https://doi.org/10.1007/s11590-023-01990-1

Bottani, E., Ferretti, G., Montanari, R., & Rinaldi, M. (2014). Analysis and optimisation of inventory management policies for perishable food products: A simulation study. *International Journal of Simulation and Process Modelling*, 119(1–2), 16–32. https://doi.org/10.1504/IJSPM.2014.061429

Bräysy, O. (2003). A reactive variable neighborhood search for the vehicle-routing problem with time windows. *INFORMS Journal on Computing*, *15*(4), 347–368. https://doi.org/10.1287/ijoc.15.4.347.24896

Brimberg, J., Salhi, S., Todosijević, R., & Urošević, D. (2023). Variable neighborhood search: The power of change and simplicity. *Computers & Operations Research*, *155*, 106221. https://doi.org/10.1016/j.cor.2023.106221

Buisman, M., Haijema, R., & Bloemhof-Ruwaard, J. (2019). Discounting and dynamic shelf life to reduce fresh food waste at retailers. *International Journal of Production Economics*, *209*, 274–284. https://doi.org/10.1016/j.ijpe.2017.07.016

Buzby, J. C., & Hyman, J. (2012). Total and per capita value of food loss in the United States. *Food Policy*, *37*(5), 561–570. https://doi.org/10.1016/j.foodpol.2012.06.002

Cachon, G. P., & Swinney, R. (2009). Purchasing, pricing, and quick response in the presence of strategic consumers. *Management Science*, *55*(3), 497–511. https://doi.org/10.1287/mnsc.1080.0948

Cannella, S., & Ciancimino, E. (2010). On the bullwhip avoidance phase: Supply chain collaboration and order smoothing. *International Journal of Production Research*, *48*(22), 6739–6776. https://doi.org/10.1080/00207540903252308

Carrefour (2022). *Tackling food wastage.* www.carrefour.com.

Chang, P. C., Chen, S. H., & Fan, C. Y. (2009). A hybrid electromagnetism-like algorithm for single machine scheduling problem. *Expert Systems with Applications*, *36*(2), 1259–1267. https://doi.org/10.1016/j.eswa.2007.11.050

Chen, B., Qu, R., Bai, R., & Laesanklang, W. (2020). A variable neighborhood search algorithm with reinforcement learning for a real-life periodic vehicle routing problem with time windows and open routes. *RAIRO - Operations Research*, *54*(5), 1467–1494. https://doi.org/10.1051/ro/2019080

Chen, X., Wu, S., Wang, X., & Li, D. (2019). Optimal pricing strategy for the perishable food supply chain. *International Journal of Production Research*, *57*(9), 2755–2768. https://doi.org/10.1080/00207543.2018.1557352

Chintapalli, P. (2015). Simultaneous pricing and inventory management of deteriorating perishable products. *Annals of Operations Research*, *229*(1), 287–301. https://doi.org/10.1007/s10479-014-1753-9

Chopard, B., Tomassini, M., Chopard, B., & Tomassini, M. (2018). Performance and limitations of metaheuristics. *An introduction to metaheuristics for optimization* (pp. 191–203). Springer.

Chung, J., & Li, D. (2014). A simulation of the impacts of dynamic price management for perishable foods on retailer performance in the presence of need-driven purchasing consumers. *Journal of the Operational Research Society*, *65*(8), 1177–1188. https://doi.org/10.1057/jors.2013.63

Cicatiello, C., Blasi, E., Giordano, C., Martella, A., & Franco, S. (2020). "If only I Could Decide": Opinions of food category managers on in-store food waste. *Sustainability*, *12*(20), 8592. https://doi.org/10.3390/su12208592

Cicatiello, C., Secondi, L., & Principato, L. (2019). Investigating consumers' perception of discounted suboptimal products at retail stores. *Resources*, *8*(3), 129. https://doi.org/10.3390/resources8030129

Davoudi, S., Stasinopoulos, P., & Shiwakoti, N. (2024). Two decades of advancements in cold supply chain logistics for reducing food waste: A review with focus on the meat industry. *Sustainability*, *16*(16), 6986. https://doi.org/10.3390/su16166986

Derringer, G., & Suich, R. (1980). Simultaneous optimization of several response variables. *Journal of Quality Technology*, *12*(4), 214–219. https://doi.org/10.1080/00224065.1980.11980968

Devika, K., Jafarian, A., & Nourbakhsh, V. (2014). Designing a sustainable closed-loop supply chain network based on triple bottom line approach: A comparison of metaheuristics hybridization techniques. *European Journal of Operational Research*, *235*(3), 594–615. https://doi.org/10.1016/j.ejor.2013.12.032

Devin, B., & Richards, C. (2018). Food waste, power, and corporate social responsibility in the Australian food supply chain. *Journal of Business Ethics*, *150*(1), 199–210. https://doi.org/10.1007/s10551-016-3181-z

Dey, S. K., Kundu, K., & Das, P. (2024). Digital technology based game-theoretic pricing strategies in a three-tier perishable food supply chain. *Annals of Operations Research.* https://link.springer.com/article/10.1007/s10479-024-06021-5

Dobslaw, F. (2010). *A parameter tuning framework for metaheuristics based on design of experiments and artificial neural networks*. In B. Brojack (Ed.), *Proceeding of the International Conference on Computer Mathematics and Natural Computing*. WASET.

Eskandarpour, M., Zegordi, S. H., & Nikbakhsh, E. (2013). A parallel variable neighborhood search for the multi-objective sustainable post-sales network design problem. *International Journal of Production Economics*, *145*(1), 117–131. https://doi.org/10.1016/j.ijpe.2012.10.013

Estrada-Moreno, A., Fikar, C., Juan, A. A., & Hirsch, P. (2019). A biased-randomized algorithm for redistribution of perishable food inventories in supermarket chains. *International Transactions in Operational Research*, *26*(6), 2077–2095. https://doi.org/10.1111/itor.12668

FAO (2023). *In brief to the state of food and agriculture 2023. Revealing the true cost of food to transform agrifood systems.* Food and Agriculture Organization of the United Nations. http://faostat.fao.org

Golden, B. L., & Skiscim, C. C. (1986). Using simulated annealing to solve routing and location problems. *Naval Research Logistics Quarterly*, *33*(2), 261–279. https://doi.org/10.1002/nav.3800330209

Govindan, K., Jafarian, A., Khodaverdi, R., & Devika, K. (2014). Two-echelon multiple-vehicle location–routing problem with time windows for optimization of sustainable supply chain network of perishable food. *International Journal of Production Economics*, *152*, 9–28. https://doi.org/10.1016/j.ijpe.2013.12.028

Govindan, K., Jafarian, A., & Nourbakhsh, V. (2015). Bi-objective integrating sustainable order allocation and sustainable supply chain network strategic design with stochastic demand using a novel robust hybrid multi-objective metaheuristic. *Computers & Operations Research*, *62*, 112–130. https://doi.org/10.1016/j.cor.2014.12.014

Govindan, K., Jafarian, A., & Nourbakhsh, V. (2019). Designing a sustainable supply chain network integrated with vehicle routing: A comparison of hybrid swarm intelligence metaheuristics. *Computers & Operations Research*, *110*, 220–235. https://doi.org/10.1016/j.cor.2018.11.013

Guide, V. D. R., Jr. & Vanwassenhove, L. N. (2001). Managing product returns for remanufacturing. *Production and Operations Management*, *10*(2), 142–155. https://doi.org/10.1111/j.1937-5956.2001.tb00075.x

Guner Goren, H., Tunali, S., & Jans, R. (2010). A review of applications of genetic algorithms in lot sizing. *Journal of Intelligent Manufacturing*, *21*(4), 575–590. https://doi.org/10.1007/s10845-008-0205-2

Guritno, A. D., Fujianti, R., & Kusumasari, D. (2015). Assessment of the supply chain factors and classification of inventory management in suppliers' level of fresh vegetables. *Agriculture and Agricultural Science Procedia*, *3*, 51–55. https://doi.org/10.1016/j.aaspro.2015.01.012

Hà, M. H., Nguyen, T. D., Duy, T. N., Pham, H. G., Do, T., & Rousseau, L. M. (2020). A new constraint programming model and a linear programming-based adaptive large neighborhood search for the vehicle routing problem with synchronization constraints. *Computers & Operations Research*, *124*, 105085. https://doi.org/10.1016/j.cor.2020.105085

Hansen, P., & Mladenovic, N. (2003). A tutorial on variable neighborhood search. *Les Cahiers du GERAD ISSN*, *711*, 2440. https://staff.fmi.uvt.ro/~daniela.zaharie/ma2017/projects/techniques/VNS/VNS_tutorial.pdf

Hansen, P., Mladenović, N., Todosijević, R., & Hanafi, S. (2017). Variable neighborhood search: Basics and variants. *EURO Journal on Computational Optimization*, *5*(3), 423–454. https://doi.org/10.1007/s13675-016-0075-x

Huang, C., Li, Y., & Yao, X. (2020). A survey of automatic parameter tuning methods for metaheuristics. *IEEE Transactions on Evolutionary Computation*, *24*(2), 201–216. https://doi.org/10.1109/TEVC.2019.2921598

Jafarian, A., Asgari, N., Mohri, S. S., Fatemi-Sadr, E., & Farahani, R. Z. (2019). The inventory-routing problem subject to vehicle failure. *Transportation Research Part E: Logistics and Transportation Review*, *126*, 254–294. https://doi.org/10.1016/j.tre.2019.04.009

Kamran, M. A., Kia, R., Goodarzian, F., & Ghasemi, P. (2023). A new vaccine supply chain network under COVID-19 conditions considering system dynamic: Artificial intelligence algorithms. *Socio-Economic Planning Sciences*, *85*, 101378. https://doi.org/10.1016/j.seps.2022.101378

Karakostas, P., & Sifaleras, A. (2022). A double-adaptive general variable neighborhood search algorithm for the solution of the traveling salesman problem. *Applied Soft Computing*, *121*, 108746. https://doi.org/10.1016/j.asoc.2022.108746

Karakostas, P., & Sifaleras, A. (2024). The pollution traveling salesman problem with refueling. *Computers & Operations Research*, *167*, 106661. https://doi.org/10.1016/j.cor.2024.106661

Karami, D. (2022). Supply chain network design using particle swarm optimization (PSO) Algorithm. *International Journal of Industrial Engineering and Operational Research*, *4*(1), 1–8. https://doi.org/10.22034/ijieor.v4i1.28

Karaoglan, I., & Altiparmak, F. (2010). A hybrid genetic algorithm for the location-routing problem with simultaneous pickup and delivery. *Computers and Industrial Engineering (CIE), 2010 40th International Conference*. IEEE.

Kayikci, Y., Demir, S., Mangla, S. K., Subramanian, N., & Koc, B. (2022). Data-driven optimal dynamic pricing strategy for reducing perishable food waste at retailers. *Journal of Cleaner Production*, *344*, 131068. https://doi.org/10.1016/j.jclepro.2022.131068

Kazimierowicz, J. (2018). Legal regulations and methods neutralising expired food products. *Journal of Ecological Engineering*, *19*(6), 217–224. https://doi.org/10.12911/22998993/89828

Kiil, K., Dreyer, H. C., Hvolby, H. H., & Chabada, L. (2018). Sustainable food supply chains: The impact of automatic replenishment in grocery stores. *Production Planning & Control*, *29*(2), 106–116. https://doi.org/10.1080/09537287.2017.1384077

Kowalska, A., & Manning, L. (2021). Using the rapid alert system for food and feed: Potential benefits and problems on data interpretation. *Critical Reviews in Food Science and Nutrition*, *61*(6), 906–919. https://doi.org/10.1080/10408398.2020.1747978

Liljestrand, K. (2017). Logistics solutions for reducing food waste. *International Journal of Physical Distribution & Logistics Management*, *47*(4), 318–339. https://doi.org/10.1108/IJPDLM-03-2016-0085

Liu, A., Zhu, Q., Xu, L., Lu, Q., & Fan, Y. (2021). Sustainable supply chain management for perishable products in emerging markets: An integrated location-inventory-routing model. *Transportation Research Part E: Logistics and Transportation Review*, *150*, 102319. https://doi.org/10.1016/j.tre.2021.102319

Liu, W., Dridi, M., Ren, J., El Hassani, A. H., & Li, S. (2023). A double-adaptive general variable neighborhood search for an unmanned electric vehicle routing and scheduling problem in green manufacturing systems. *Engineering Applications of Artificial Intelligence*, *126*, 107113. https://doi.org/10.1016/j.engappai.2023.107113

Lundqvist, J. (2009). Losses and waste in the global crisis. *Reviews in Environmental Science and Bio/Technology*, *8*(2), 121–123. https://doi.org/10.1007/s11157-009-9152-4

Lütke Entrup, M., Günther, H.-O., Van Beek, P., Grunow, M., & Seiler, T. (2005). Mixed-Integer Linear Programming approaches to shelf-life-integrated planning and scheduling in yoghurt production. *International Journal of Production Research*, *43*(23), 5071–5100. https://doi.org/10.1080/00207540500161068

Lysova, N., Solari, F., Bottani, E., & Montanari, R. (2024). Impact of the discount policies on the purchasing behaviour of perishable items. *IFAC-PapersOnLine*, *58*(19), 433–438. https://doi.org/10.1016/j.ifacol.2024.09.250

Martins, C., & Pato, M. (2024). Decomposition heuristics for multiobjective problems. The Food bank network redesign case. *International Journal of Production Economics*, *268*, 109121. https://doi.org/10.1016/j.ijpe.2023.109121

Mladenović, N., & Hansen, P. (1997). Variable neighborhood search. *Computers & Operations Research*, *24*(11), 1097–1100. https://doi.org/10.1016/S0305-0548(97)00031-2

Mogale, D., Dolgui, A., Kandhway, R., Kumar, S. K., & Tiwari, M. K. (2017). A multi-period inventory transportation model for tactical planning of food grain supply chain. *Computers & Industrial Engineering*, *110*, 379–394. https://doi.org/10.1016/j.cie.2017.06.008

Montero, E., Riff, M.-C., & Neveu, B. (2014). A beginner's guide to tuning methods. *Applied Soft Computing*, *17*, 39–51. https://doi.org/10.1016/j.asoc.2013.12.017

Montgomery, D. C. (2017). *Design and analysis of experiments*. John wiley & sons.

Mostajabdaveh, M., Gutjahr, W. J., & Sibel Salman, F. (2019). Inequity-averse shelter location for disaster preparedness. *IISE Transactions*, *51*(8), 809–829. https://doi.org/10.1080/24725854.2018.1496372

Mousavi, S. M., Bahreininejad, A., Musa, S. N., & Yusof, F. (2017). A modified particle swarm optimization for solving the integrated location and inventory control problems in a two-echelon supply chain network. *Journal of Intelligent Manufacturing*, *28*(1), 191–206. https://doi.org/10.1007/s10845-014-0970-z

Nguyen, V. P., Prins, C., & Prodhon, C. (2012). Solving the two-echelon location routing problem by a GRASP reinforced by a learning process and path relinking. *European Journal of Operational Research*, *216*(1), 113–126. https://doi.org/10.1016/j.ejor.2011.07.030

Nicolai, R. P., Dekker, R., Piersma, N., van Oortmarssen, G. J. (2004). Automated response surface methodology for stochastic optimization models with unknown variance. *Proceedings of the 2004 Winter Simulation Conference, 2004*. IEEE.

Nunes Bezerra, S., Souza, M. J. F., de Souza, S. R., & Nazário Coelho, V. (2019, October 4–7, 2018). *A VNS-based algorithm with adaptive local search for solving the multi-depot vehicle routing problem. Variable Neighborhood Search* [Paper presentation].6th International Conference, ICVNS 2018, Revised Selected Papers 6, Sithonia, Greece.

Qiu, Y., Qiao, J., & Pardalos, P. M. (2019). Optimal production, replenishment, delivery, routing and inventory management policies for products with perishable inventory. *Omega*, *82*, 193–204. https://doi.org/10.1016/j.omega.2018.01.006

Rabbani, M., Tohidi Fard, M., Partovi, M., & Farrokhi-Asl, H. (2018). Solving a multi-depot vehicle routing problem with time windows and fuzzy demands using metaheuristic algorithms in home health care services.

Journal of Decisions and Operations Research, *3*(2), 114–127. https://doi.org/10.22105/dmor.2018.65276

Rabiee, M., Zandieh, M., & Jafarian, A. (2012). Scheduling of a no-wait two-machine flow shop with sequence-dependent setup times and probable rework using robust meta-heuristics. *International Journal of Production Research*, *50*(24), 7428–7446. https://doi.org/10.1080/00207543.2011.652747

Riesenegger, L., & Hübner, A. (2022). Reducing food waste at retail stores—an explorative study. *Sustainability*, *14*(5), 2494. https://doi.org/10.3390/su14052494

Ross, A., & Jayaraman, V. (2008). An evaluation of new heuristics for the location of cross-docks distribution centers in supply chain network design. *Computers & Industrial Engineering*, *55*(1), 64–79. https://doi.org/10.1016/j.cie.2007.12.001

Rout, C., Paul, A., Kumar, R. S., Chakraborty, D., & Goswami, A. (2021). Integrated optimization of inventory, replenishment and vehicle routing for a sustainable supply chain under carbon emission regulations. *Journal of Cleaner Production*, *316*, 128256. https://doi.org/10.1016/j.jclepro.2021.128256

Sadeghi, J., Sadeghi, S., & Niaki, S. T. A. (2014). Optimizing a hybrid vendor-managed inventory and transportation problem with fuzzy demand: An improved particle swarm optimization algorithm. *Information Sciences*, *272*, 126–144. https://doi.org/10.1016/j.ins.2014.02.075

Saremi, A., ElMekkawy, T., & Wang, G. (2007). Tuning the parameters of a memetic algorithm to solve vehicle routing problem with backhauls using design of experiments. *International Journal of Operations Research*, *4*(4), 206–219. https://www.sfu.ca/~gwa5/index_files/MA-Ali-07.pdf

Sarvary, M., & Padmanabhan, V. (2001). The informational role of manufacturer returns policies: How they can help in learning the demand. *Marketing Letters*, *12*(4), 341–350. https://doi.org/10.1023/A:1012224422616

Schneider, M., Stenger, A., & Hof, J. (2015). An adaptive VNS algorithm for vehicle routing problems with intermediate stops. *Or Spectrum*, *37*(2), 353–387. https://doi.org/10.1007/s00291-014-0376-5

Shadkam, E. (2022). Parameter setting of meta-heuristic algorithms: A new hybrid method based on DEA and RSM. *Environmental Science and Pollution Research International*, *29*(15), 22404–22426. https://doi.org/10.1007/s11356-021-17364-y

Shi, W., Wang, N., Zhang, M., & Jiang, B. (2023). A bi-objective pollution routing optimisation problem with decentralised cooperation and split delivery. *IEEE Transactions on Intelligent Transportation Systems*, *24*(11), 12357–12371. https://doi.org/10.1109/TITS.2023.3293507

Soleimani, H., & Kannan, G. (2015). A hybrid particle swarm optimization and genetic algorithm for closed-loop supply chain network design in large-scale networks. *Applied Mathematical Modelling*, *39*(14), 3990–4012. https://doi.org/10.1016/j.apm.2014.12.016

Somkun, P. N. (2017). Stochastic mathematical model for food waste reduction in a two-level supply chain for highly perishable products. *International Journal of Supply Chain Management*, *6*(1), 165–171. https://doi.org/10.59160/ijscm.v6i1.1585

Sze, J. F., Salhi, S., & Wassan, N. (2024). An adaptive variable neighbourhood search approach for the dynamic vehicle routing problem. *Computers & Operations Research*, *164*, 106531. https://doi.org/10.1016/j.cor.2024.106531

Talbi, E. G. (2021). Machine learning into metaheuristics: A survey and taxonomy. *ACM Computing Surveys (CSUR)*, *54*(6), 1–32. https://dl.acm.org/doi/abs/10.1145/3459664

Tarantilis, C., & Kiranoudis, C. (2001). A meta-heuristic algorithm for the efficient distribution of perishable foods. *Journal of Food Engineering*, *50*(1), 1–9. https://doi.org/10.1016/S0260-8774(00)00187-4

Teller, C., Holweg, C., Reiner, G., & Kotzab, H. (2018). Retail store operations and food waste. *Journal of Cleaner Production*, *185*, 981–997. https://doi.org/10.1016/j.jclepro.2018.02.280

Tohidifard, M., Tavakkoli-Moghaddam, R., Navazi, F., & Partovi, M. (2018). A multi-depot home care routing problem with time windows and fuzzy demands solving by particle swarm optimization and genetic algorithm. *IFAC-PapersOnLine*, *51*(11), 358–363. https://doi.org/10.1016/j.ifacol.2018.08.318

UNEP Food Waste Index Report. (2021). *UNEP food waste index report.]*. http://www.ipsnews.net/2021/03/how-households-increased-food-waste-is-feeding-climate-change170488/

Vahdani, B., & Zandieh, M. (2010). Scheduling trucks in cross-docking systems: Robust meta-heuristics. *Computers & Industrial Engineering*, *58*(1), 12–24. https://doi.org/10.1016/j.cie.2009.06.006

Wang, G. (2023). Designing a stochastic supply chain network: An error-bound-based heuristic. *Computers & Industrial Engineering*, *184*, 109570. https://doi.org/10.1016/j.cie.2023.109570

Wang, N., Jiang, Q., Jiang, B., & He, Z. (2022). *Enterprises' green growth model and value chain reconstruction*. Springer Books.

Wang, X., Rodrigues, V. S., & Demir, E. (2019). Managing your supply chain pantry: Food waste mitigation through inventory control. *IEEE Engineering Management Review*, *47*(2), 97–102. https://doi.org/10.1109/EMR.2019.2915064

Woeginger, G. J. (2003, March 5–9). *Exact algorithms for NP-hard problems: A survey* [Paper presentation]. Combinatorial Optimization—Eureka, You Shrink! Papers Dedicated to Jack Edmonds 5th International Workshop Aussois, 2001 Revised Papers, France.

Yang, S., Xiao, Y., & Kuo, Y. H. (2017). The supply chain design for perishable food with stochastic demand. *Sustainability*, *9*(7), 1195. https://doi.org/10.3390/su9071195

Yang, X. S., Deb, S., Loomes, M., & Karamanoglu, M. (2013). A framework for self-tuning optimization algorithm. *Neural Computing and Applications*, *23*(7–8), 2051–2057. https://doi.org/10.1007/s00521-013-1498-4

Zandieh, M., & Gholami, M. (2009). An immune algorithm for scheduling a hybrid flow shop with sequence-dependent setup times and machines with random breakdowns. *International Journal of Production Research*, *47*(24), 6999–7027. https://doi.org/10.1080/00207540802400636

Zanoni, S., & Zavanella, L. (2012). Chilled or frozen? Decision strategies for sustainable food supply chains. *International Journal of Production Economics*, *140*(2), 731–736. https://doi.org/10.1016/j.ijpe.2011.04.028

Zhou, L., Zhang, D., Li, S., & Luo, X. (2023). An integrated optimization model of green supply chain network design with inventory management. *Sustainability*, *15*(16), 12583. https://doi.org/10.3390/su151612583

Zhu, L., & Hu, D. (2019). Study on the vehicle routing problem considering congestion and emission factors. *International Journal of Production Research*, *57*(19), 6115–6129. https://doi.org/10.1080/00207543.2018.1533260