International Journal of Production Research, Volume 61, Issue 5, March 2023, pp. 1455-1478
DOI: 10.1080/00207543.2022.2037777

Designing a sustainable freight transportation network with cross-docks

D. G. Mogale[a*], Arijit De[b], Abhijeet Ghadge[c], Manoj Kumar Tiwari[d]

[a]*Logistics and Operations Management Section, Cardiff Business School, Cardiff University, Cardiff, UK.*
[b] *Alliance Manchester Business School, University of Manchester, UK*
[c]*Centre for Logistics and Supply Chain Management, School of Management, Cranfield University, MK43 0AL, UK*
[d]*National Institute of Industrial Engineering (NITIE), Mumbai, India*

Article in ***International Journal of Production Research*, 2022**

Abstract

This study aims to develop a sustainable freight transportation network considering capacitated cross-docks for minimising the overall supply chain costs, including carbon emission cost. The problem is inspired by a major retail company based in India, which would like to expand its product portfolio in the new region. A mathematical model is developed to minimise total costs encompassing transportation cost, pipeline and retailers inventory cost, fixed cost of cross-dock and carbon emission costs. The deterministic time dependant demand, multiple products and multiple sourcing and distribution are some of the challenges faced by the retail industry. A two-level self-adaptive variable neighbourhood search algorithm is applied to solve a computationally complex problem. The results based on a two-level self-adaptive variable neighbourhood search algorithm are compared with the variable neighbourhood search algorithm to test the robustness of the developed model. Results reveal that an increase in retailers over suppliers significantly influences the number of open cross-docks. A multiple-case scenario approach captures the implications of varying capacity on the number of open cross-docks; thus, supporting the freight distribution managers in making sustainability-driven decisions.

Keywords:

Sustainable freight transportation; Cross-docking; Transportation; Inventory; Mathematical modelling, Variable Neighbourhood Search (VNS)

1. Introduction

Globalisation and rapid economic growth have created an increasing demand for the logistics sector leading to a surge in the carbon emission levels (Shankar et al., 2019, Kumar and Anbanandam, 2020; Yazdani et al., 2020). The freight transportation has been imposing several negative externalities on the environment such as air pollution, traffic congestion, noise pollution and accidents (Demir et al., 2019; Yazdani et al., 2020). With regards to the social dimension of sustainability, globally 1.35 million people lose their lives every year or left with severe injuries due to road traffic accidents (WHO, 2020). Sustainable freight transportation helps to reduce energy consumption and greenhouse gas emission through the use of low-carbon fuel, electric mobility and adoption of environmental standards and green practices resulting in the substantial savings in operational costs and carbon footprint regulations (Pathak et al., 2019; Goswami et al., 2020; Mahapatra et al., 2020). It also provides several social benefits related to employment, education, health and balanced economic development (de Campos et al., 2019; Mogale et al., 2020). The research on the sustainable freight transportation has received ever-increasing attention from researchers and practitioners due to changes in governmental policies, technological developments, climate change and consumers pressure in the recent times (SteadieSeifi et al., 2014; Dente and Tavasszy, 2017; Kumar and Anbanandam, 2019, Ghadge et al., 2020).

Cross-docking is recognised as the most appealing sustainable approach for freight transportation (Dulebenets 2018; Rezaei and Kheirkhah 2018; Kiani Mavi et al., 2020), and has received significant research interest in recent times (e.g., Abad et al., 2018; Chargui et al., 2019; Tirkolaee et al., 2020, Urzúa-Morales et al., 2020, Pan et al., 2021). Cross-docking efficiently cuts the costs of retrieval and storage of products up to 70% compared with traditional warehouses (Vahdani and Zandieh, 2010, Abad et al., 2018, Vahdani et al., 2019). The inventory stock does not stay more than 24 hours at the cross-docking facility (Ladier and Alpan, 2016), which results into the reduction of energy consumption for keeping and managing the stocks (Dulebenets 2018, Vahdani, 2019). It synchronises the inbound and the outbound transportation systems and avoids inappropriate procedures (transportation loops and Less-Than-Truckload (LTL)) shipments, reduces operational and transportation cost, storage time, delivery time, reduces risks of product damage and obsolescence, and carbon emissions (Van Belle et al., 2012; Moghadam et al., 2014; Yu et al., 2016; Dulebenets, 2018; Rezaei and Kheirkhah, 2018; Gaudioso et al., 2021). Given significant impact and benefits of freight transportation with cross-docking to sustainability, it is imperative to re-design freight transportation by considering cross-docking, sustainability, and traditional factors (Kumar et al., 2019, Vahdani et al., 2019, Kumar and Anbanandam, 2020, Tirkolaee et al., 2020).

This research study is motivated by a major Indian retail company, which has stores located mainly in the southern region of India like Bangalore, Chennai, and Hyderabad. The company is looking for expansion in the northern region due to the entry of external/global players in the Indian market to target more customers. A large number of retail shops, high product variety, continuously changing lifestyle and logistics requirements make the Indian retail sector more complex and dynamic (Gawankar et al., 2020). This sector contributes 10% to India's Gross Domestic Product (GDP) and employs approximate 8% of its workforce (Khanna, 2020). Many big retail players are utilising the cross-docking facility worldwide, and the company under study would like to follow a similar approach. Hence, they are interested in optimising freight supply chain cost by establishing a cross-dock facility instead of a warehouse to meet the demand of customers from the northern region. The transport sector in India contributes nearly 10% of total CO_2 Emissions (Garg et al., 2017). The Intended Nationally Determined Contributions (INDC) of India considers transport as a key sector to reduce the CO_2 emissions by 30-35% from 2005 to 2030 (UNFCCC, 2015). To achieve this carbon footprint goal, the central government in India is setting up various emission norms, and regulatory bodies make sure that the freight transport companies meet emission compliance (Goswami et al., 2020). Following the need for considering cross-docking in sustainable supply chains, this research aims to develop a sustainable freight transportation network by designing a capacitated cross-docking system for minimising the overall retail supply chain costs.

The computation of the proposed sustainable freight transportation problem becomes complicated as the problem size increases exponentially with the increase in number of suppliers, cross-docks and retailers. Contemporary optimization techniques such as exact heuristics require the mathematical model to be continuous and linear and furthermore, demands substantial computational effort in terms time complexity for solving purpose (Govindan et al. 2019, De at al. 2019a, Maiyar et al. 2019). Thereby, highlighting that exact solution methods are insufficient in solving large instances of real-world sustainable freight transportation problem (De et al. 2020). Yang et al. 2015 employed a variable neighborhood search (VNS) heuristic for dealing with medium and large size problems as it becomes increasingly difficult to solve problem instances exact methods. VNS algorithm is used with the motivation of obtaining near-optimal solution for the problem with a better computational efficiency.

Several researchers have implemented meta-heuristic such as VNS algorithm to address complex problems such as packing problems (M'Hallah et al. 2013), location routing problems (Jarboui et al. 2013), etc. VNS has also been used as a methodology to solve inventory routing problem (Hansen et al. 2010, Hansen et al. 2001). Furthermore, VNS algorithm gives superior results in less computational time for a variety of problems when compared with several other benchmark algorithms (Govindan et al., 2019; Gruler et al., 2020). VNS-based algorithm has proved to be successful in solving a variety of combinatorial problem (Salhi et al. 2014) and success of two-level general VNS in addressing traveling salesman problem is well documented (Mladenović et al. 2014). Hence, for solving the sustainable freight transportation problem, VNS algorithm and Two-Level Self-Adaptive Variable Neighbourhood Search (TLSAVNS) algorithm have been adopted in this work. Evidently, there is a methodological research gap within the literature pertaining to the lack of exploring the potential of VNS algorithm for solving sustainable freight transportation problem. Thus, this research work explores the application of VNS and TLSAVNS algorithms and validates its pertinency for the aforementioned problem, while considering large-size real-world problem instances.

The contribution of the paper is in two folds. Firstly, this paper attempts to develop a decision support model for the capacitated cross-docking system problem in the retail industry, to reduce overall supply chain costs, including the cost associated with carbon emissions. The complex problem considers several suppliers, multiple locations of cross-docks and retailers. Furthermore, the developed mathematical model considers the scenario of multiple product types and time-dependent demands. Secondly, the study presents the application of the Two-Level Self-Adaptive Variable Neighbourhood Search (TLSAVNS) algorithm and justifies its applicability for the capacitated cross-docking system problem. Considering several large-size problems, the superiority of the TLSAVNS algorithm over the VNS algorithm is established from the perspective of total cost and computational time.

The rest of the paper is organised as follows: In section 2, a brief literature background of sustainable freight transportation and cross-docking, and VNS algorithm along with research gap and contributions is provided. Problem description and development of the proposed mathematical model are presented in section 3. Section 4 explains the algorithms used for solving a capacitated cross-docking system. Experimental computation, analysis of results, and the contribution to theory and practice are presented in section 5. Finally, section 6 concludes with key findings and limitations of the research.

2. Literature review

In this section, background literature on sustainable freight transportation and cross-docking, and VNS are discussed.

2.1. Sustainable freight transportation and cross-docking

Bertazzi et al. (2016) addressed a freight (inventory) transportation problem of the first kind with the delivery of a single product via outsourced vehicles. However, the consideration of variable transportation costs and carbon emission costs were not considered in their model. Similarly, Lee et al. (2016) designed a synchronised supply chain by minimising the expected inventory cost for the freight transportation problem but lacked the consideration of sustainability-related costs. Recently, Dulebenets (2018) addressed the sustainable truck scheduling problem at a cross-docking facility to minimise total truck service cost. The environmental sustainability aspect was missing in this study. Similarly, several researchers have incorporated freight transportation models into their frameworks (e.g., Berman and Wang, 2006; Konur and Schaefer, 2014; Schaefer and Konur, 2015; Zhao et al., 2016; Mogale et al., 2018, Urzúa-Morales et al., 2020). However, consideration of the carbon emission costs, and other cost elements were found to be lacking in the extant literature (Abouee-Mehrizi et al., 2014, Dulebenets 2018, Kumar and Anbanandam, 2019, Kumar et al., 2019). A cross-dock helps to achieve economies in transportation through the consolidation of inventory for a shorter period (Benrqya, 2019; Dulebenets et al., 2019; Wu et al. (2015). Furthermore, Chen et al. (2016), Goodarzi et al. (2020), Goodarzi and Zegordi (2016) and Maknoon and Laporte (2017) solved a cross-dock location-routing problem to minimise the operating costs. These studies add to the existing literature in cross-docking logistics with vehicle routing and allocation problems. Interested readers can refer to the review articles on cross-docking by Van Belle et al. (2012), Ladier and Alpan (2016) and Kiani Mavi et al. (2020) for further details.

2.2. Variable Neighbourhood Search (VNS)

Initially developed by Mladenović and Hansen (1997), Variable Neighbourhood Search (VNS) algorithm is a generic local search methodology employed for solving combinatorial optimisation problems (Xiao et al., 2014). It is a combinatorial and global optimisation algorithm with a systematic change in the neighbourhood search (Hansen and Mladenović, 2014). Successful application of the VNS algorithm has been observed for a variety of problems, and the VNS metaheuristic also provides robust results in less computational time, when compared to other benchmark algorithms like Genetic Algorithm (GA), Ant Colony Optimization (ACO), Simulated Annealing (SA) and Tabu Search (TS) (Govindan et al., 2019; Gruler et al., 2020). The reason behind the success of VNS is the notion of expanding the neighbourhood, when the search is stuck at a local optimum (Menendez et al., 2017; Govindan et al., 2019). The VNS algorithm overcomes the local optima through the exploration of a solution space exhaustively and employing an operator (named '*shaking'*) to improve the solution by utilising '*local search'* (Alguwaizani et al., 2011). Application of VNS-based algorithm to deal with a sustainable capacitated cross-docking problem is new to the broad domain of the literature. The VNS algorithm exploits the idea of systematic neighbourhood change in two phases - the decline phase (for finding local optimum) and the perturbation phase (to get out from the local minimum) (Hansen et al., 2010). The systematic change of the neighbourhood in the search procedure does not follow a single trajectory but, instead, explores increasingly distant neighbours of the demanded solution, moving from one local optimum to another, only if there is a visible improvement on the way.

2.3. Research gaps

The existing studies in supply chain management broadly revolve around supply chain cost, carbon tax, cost of carbon emissions and the impact of the carbon footprint on the environment. A summary of the key relevant studies showcasing the novelty of our work is provided in Table 1. Multiple suppliers (Goodarzi et al., 2020) and echelons (De et al., 2020) are considered in the previous studies; however, a limited number of scholars embedded the cross-dock and multi-period features in their developed model (e.g., Chen et al., 2016, Küçükoğlu and Öztürk 2017, Rezaei and Kheirkhah 2017). Several scholars worked on vehicle routing and scheduling at the cross-docking facility. To the best of our knowledge, no studies have considered freight transportation, cross-docking, and sustainability simultaneously. The sustainability issue has been mostly overlooked in the cross-docking context in the extant literature (Abad et al., 2018, Dulebenets 2018, Chargui et al., 2019, Goodarzi et al. 2020). Research in freight transportation considering capacitated cross-docking as an approach for enhancing sustainability is lacking in the existing literature (evident from Table 1). The majority of past studies are from the developed regions/countries (The US and

Europe) and their freight transportation network and government policies are different from developing regions/countries. Lack of efficient and reliable transportation and logistics infrastructure, poor service quality, greater use of simpler technology and older equipment (e.g., vehicles), cutting transportation costs and improving supply chain visibility are some of the major challenges in the developing world. Hence, there is a huge scope for researchers to address the sustainable freight transportation issues from developing regions like Asia (Kumar and Anbanandam, 2019). With regards to decisions making, location-allocation and transportation decisions are mostly explored in the extant literature (Wu et al., 2015), whereas pipeline and retailers' inventory are hardly observed in the literature (Bertazzi et al., 2016). Interestingly, a comprehensive model for a sustainable cross-docking system (as a freight transportation system), considering multiple economic and environmental factors is lacking in the current body of literature. With regards to the solution approach, none of the relevant studies discussed in the literature explored the potential of VNS algorithm to solve the sustainable freight transportation problem. This is an evident research gap, and this study attempts to address it. A mathematical formulation for the capacitated cross-docking system is developed and solved using VNS based metaheuristic approach.

Table 1. Summary of the features of key relevant studies

Abbreviations:

Model features*: MS: Multi-suppliers, CD: Cross-dock, ME: Multi-echelon, MP: Multi-products, CEC: Carbon emission cost, DIS: Direct and indirect shipment*

Decisions: *Loc: Location, Alloc: Allocation, PI: Pipeline Inventory, RI: Retailer Inventory and Trans: Transportation*

Solution methods*: NICE: Nested Integrated Cross-Entropy method, GA: Genetic algorithm, SLPSO: Self-learning particle swarm optimisation, BBO: Biogeography-based optimisation, DSPEA: Delayed Start Parallel Evolutionary* Algorithm, SA: *Simulated Annealing, TLSAVNS: Two-level self-adaptive variable neighbourhood search, MO-VNSSA: Multi-Objective Variable Neighbourhood Search hybridized with Simulated Annealing, MO-VNSTS: Variable Neighbourhood Search VNS hybridized with Tabu Search, COA: Cuckoo Optimization Algorithm, MOSA: Multi-objective Simulated Annealing Algorithm*

Research study	**Model features**						**Decisions**					**Country/ Region**	**Solution methods**
	MS	CD	ME	MP	CEC	DIS	Loc	Alloc	PI	RI	Trans		
Goodarzi et al. (2020)	✓	✓				✓	✓				✓	Middle East	Lagrangian relaxation algorithm
Fahimnia et al. (2015)			✓	✓			✓			✓	✓	Australia	NICE method
Diabat and Al-Salem (2015)			✓		✓		✓	✓			✓	-	GA and GAMS
De et al. (2020)			✓	✓				✓		✓	✓	-	Problem-based heuristic
Chen et al. (2016)	✓	✓		✓				✓			✓	-	SLPSO
Jin et al. (2014)	✓		✓		✓						✓	USA	CPLEX
Goodarzi and Zegordi (2016)	✓	✓					✓				✓	-	BBO, PSO and GAMS
Ding et al. (2016)			✓		✓						✓	-	Game approach
Küçükoğlu and Öztürk (2017)	✓	✓	✓	✓							✓	-	GA and Gurobi solver
Dulebenets et al. (2019)		✓						✓				-	DSPEA
Chargui et al. (2019)		✓						✓				-	MO-VNSSA and MO-VNSTS
Rezaei and Kheirkhah (2017)	✓	✓	✓	✓			✓	✓			✓	-	GAMS
Liotta et al. (2015)	✓		✓	✓	✓			✓			✓	Europe	CPLEX
Maiyar and Thakkar (2019)	✓		✓		✓		✓				✓	India	PSODE
Mogale et al. (2020)	✓		✓		✓		✓	✓			✓	India	GLNPSO and PSO
Rezaei and Kheirkhah (2018)	✓	✓	✓			✓	✓	✓			✓	-	COA
Tirkolaee et al. (2020)	✓				✓			✓			✓	Iran	MOSA and NSGA-II
Wu et al. (2015)			✓				✓	✓			✓	-	Greedy Heuristic
Bertazzi et al. (2016)			✓						✓		✓	-	Min–Max Matheuristic
Lee et al. (2016)	✓					✓			✓		✓	China	GA and SA based heuristics
Current study	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	**India**	**TLSAVNS and VNS**

3. Problem description and mathematical model

The model is developed following a realistic scenario of the retail company based in India, where inventory is shipped from a supplier *p* to retailer *r* via cross-dock *q,* as shown in Figure 1. The study considers a three-echelon supply chain network model to satisfy the deterministic time-dependent demand. A cross-docking system not only helps to regulate the distribution but also avoids excess storage cost (Gaudioso et al., 2021). Currently, the organisation is covering the east and south regions of India and are looking for the expansion of business in the northern region. Thus, different production/supplier locations and cross-docks are identified and are available to satisfy a deterministic demand in the northern region. The company is not interested in the warehouse-based system due to the huge initial investment. Retailers have the option of placing the order through cross-dock or direct purchase from suppliers bypassing cross-docks. Multiple products with similar characteristics (such as weight, size, quantity) are grouped as a consignment for shipping. The model considers universal time ‘periods', for assessing quantities per unit time, where the period can be in days, weeks or months t_{pq}, t_{qr} and t_q signifies the ratio of transportation time taken to a length of period. *P* represents the set of suppliers, *R* depicts the set of retailers, *Q* represents the set of cross-docks, and *S* depicts the set of products. The model considers the following parameters and variables.

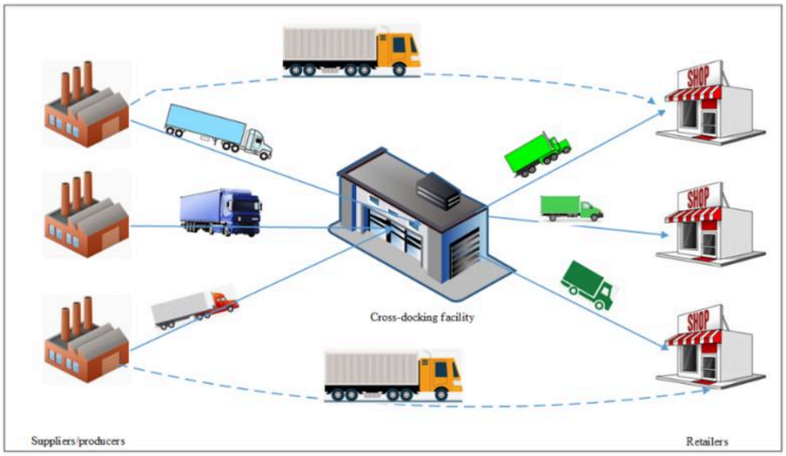

Figure 1. Representation of a retail cross-docking system.

Figure 1 Alt Text: Supply chain network showing suppliers, cross-docks and retailers along with different vehicles movement

Indices

$p = 1, 2\ldots P$ p is the index of the suppliers

$q = 1, 2\ldots Q$ q is the index of the cross-docks

$r = 1, 2\ldots R$ r is the index of the retailers

$s = 1, 2\ldots S$ s is the index of the product types

Inbound Parameters

c_{pq} Transportation cost of one truckload of products from supplier p to cross-dock q

t_{pq} The time required to transfer products from supplier p to cross-dock q

d_{pq} Total distance travelled from supplier p to cross-dock q

A_{ps} Available supply of product type s at supplier p

Outbound Parameters

c_{qr} Transportation cost of one truckload of products from cross-dock q to retailer r

t_{qr} The time required to transfer products from cross-dock q to retailer r

d_{qr} Total distance travelled from cross-dock q to retailer r

Parameters related to cross-docking system

α Total capacity of a truck

t_q Time required to transfer products from inbound to outbound door at cross-dock q

FC_q Fixed cost required to establish cross-dock q

β_q Capacity of cross-dock q

Parameters related to a direct transfer

c_{pr} Direct transportation cost of one truckload of products from supplier p to retailer r.

t_{pr} Time required to transfer products from supplier p to retailer r

d_{pr} Total distance travelled from supplier p to retailer r.

Parameters related to a product

E_{rs} Demand of product type s required by retailer r

g_s Truck capacity required for one unit of product type s

m_s Cost of inventory for one unit of product type s per period

v_s Weight of a single unit of product type s

F_s Factor to adjust product type s into standard units of supplier/cross-dock capacity

Parameters related to Carbon Emission Cost

ω Carbon emission cost per gram of CO_2 evolved

δ Carbon emissions per weight distance of vehicle (gm. /metric ton-km)

Decision Variables

X_q $\begin{cases} 1 & \text{If cross dock } q \text{ is open} \\ 0 & \text{Otherwise} \end{cases}$

Y_{prsq} $\begin{cases} 1 & \text{If supplier } p \text{ transfer the product } s \text{ to retailer } r \text{ via cross dock } q \\ 0 & \text{Otherwise} \end{cases}$

Z_{prs} $\begin{cases} 1 & \text{If supplier } p \text{ directly transfer the product } s \text{ to retailer } r \\ 0 & \text{Otherwise} \end{cases}$

K_{prsq} Units of product type s supplied by supplier p and shipped via cross-dock q to retailer r

K_{prs} Units of product type s directly supplied by supplier p to retailer r

N_{pq} The number of trucks utilised for shipment between the supplier p and cross-dock q

N_{qr} The number of trucks utilised for shipment between the cross-dock q and retailer r

N_{pr} The number of trucks utilised for direct shipment between the supplier p and retailer r

The objective of the model is to minimise the total (supply chain) cost considering multiple costs associated with the network such as pipeline and retailer inventory cost, transportation cost, fixed cost of cross-dock and cost incurred for the carbon emissions. To achieve the set objective, the following decisions must be made:

- *Location decision* - To decide the number of locations of cross-docks.
- *Allocation decision* - To decide whether to choose direct transportation of goods or via the cross-dock.
- *Pipeline and retailer inventory, and transportation decision* - To determine the pipeline and retailer inventory, frequency of inbound and outbound shipments from a cross-dock

The following costs are considered in the cross-docking system model: transportation cost, pipeline inventory cost, retailer inventory cost, fixed cost associated with each cross-dock and carbon emission cost. The transportation cost is assumed to follow the linear behaviour concerning the number of products shipped (Berman and Wang, 2006). Inbound transportation cost c_{pq}, outbound transportation cost c_{qr} and direct transfer cost c_{pr} are considered separately. Likewise, inbound transportation time t_{pq}, outbound transportation time t_{qr} and t_{pr} direct transfer time are considered separately.

The objective of the model is to minimise the total cost and is formulated as follows:

***Minimize* Total cost** = Total Transportation cost (TTC) + Total inventory Cost (TIC) + Fixed cost of cross-dock (FCCD) + Carbon emission cost (CEC)

To define the total transportation cost (TTC), the following variables are considered:

The inbound shipment frequency for each supplier *p* and cross-dock *q* is calculated as

$$N_{pq} = \sum_{r,s} \frac{g_s K_{prsq}}{\alpha} \qquad (1)$$

Moreover, the total cost of inbound transportation is given by equation (2)

$$ITC = \sum_{p,q} N_{pq} c_{pq} \qquad (2)$$

Similarly, for every cross-dock *q* and retailer *r,* the outbound shipment frequency is given by

$$N_{qr} = \sum_{p,s} \frac{g_s K_{prsq}}{\alpha} \qquad (3)$$

The total cost of outbound transportation is given by the following equation (4),

$$OTC = \sum_{q,r} N_{qr} c_{qr} \qquad (4)$$

In case of direct transfer, shipment frequency for each supplier *p* to retailer *r* is determined as

$$N_{pr} = \sum_{s} \frac{g_s K_{prs}}{\alpha} \qquad (5)$$

The total cost of direct transportation is calculated using below equation (6).

$$\text{DTC} = \sum_{p,r} N_{pr} c_{pr} \qquad (6)$$

Hence, the transportation cost is the sum of the inbound, outbound and direct cost of transportation

Total Transportation cost (TTC) = *Inbound Transportation cost (ITC) + Outbound Transportation cost (OTC)+ Direct Transportation Cost (DTC)*

TTC = ITC + OTC + DTC

TTC = $\sum_{p,q}\left(\sum_{r,s}\frac{g_s}{\alpha}K_{prsq}\right)c_{pq}+\sum_{q,r}\left(\sum_{p,s}\frac{g_s}{\alpha}K_{prsq}\right)c_{qr}+\sum_{p,r}\left(\sum_{s}\frac{g_s}{\alpha}K_{prs}\right)c_{pr}$ (7)

For all routes from supplier *p* to retailer *r* via cross-dock *q,* the total transportation time is given as

$(t_{pq}+t_{qr}+t_{q})$ (8)

Similarly, transportation time for direct route from supplier *p* to retailer *r* is given by t_{pr} .

Total inventory cost (TIC) = Pipeline inventory cost (PIC) + Retailer Inventory cost (RIC)

Therefore, the **Pipeline inventory cost (PIC)**

= $\sum_{p,r,s,q} m_s\left[K_{prsq}(t_{pq}+t_{qr}+t_{q})+K_{prs}t_{pr}\right]$ (9)

To calculate the inventory cost at the retailers (equation 10), the number of products shipped from cross-dock *q* to retailer *r* and directly transferred product from supplier *p* to retailer *r* are considered to be $\sum_{p,s}\left(K_{prsq}+K_{prs}\right)$. Cross-dock *q* transfers these products in N_{qr} shipments to retailer *r* and supplier transfers it into N_{pr} . With the help of these two entities, we can determine the total number of products directly shipped from supplier *p* to retailer *r* and from cross-dock *q* to retailer *r* in each delivery is $\left(\sum_{p,s}K_{prsq}\Big/N_{qr}\right)+\left(\sum_{p,s}K_{prs}\Big/N_{pr}\right)$. Due to the deterministic nature of the demand and existence of the linear relationship between inventory and demand, the average retailer inventory cost can be calculated (Berman and Wang, 2006).

Average retailer inventory Cost (RIC) = $\sum_{p,r,q}\frac{\sum_{p,s}m_s\left(N_{pr}K_{prsq}+N_{qr}K_{prs}\right)}{2\left(N_{qr}*N_{pr}\right)}$ (10)

Fixed cost of cross-dock (FCCD) = $\sum_{q}X_qFC_q$ (11)

The total distance covered by the trucks is the sum of the inbound, outbound and direct distance travelled $(d_{pq}+d_{qr}+d_{pr})$. The carbon emission cost is assumed to be proportional to the weight of the load, type of fuel used, and distance travelled while carrying the load.

Carbon emission cost (CEC) = $\omega\delta\sum_{p,r,s,d}(d_{pq}+d_{qr})(K_{prsq}v_s)+(d_{pr}K_{prs}v_s)$ (12)

Equation (7) can be expressed in the following way.

TTC = $\sum_{p,r,s,q}\left[\frac{g_s}{\alpha}\left(c_{pq}+c_{qr}\right)\right]K_{prsq}+\left[\frac{g_s}{\alpha}\left(c_{pr}\right)K_{prs}\right]$

Where $U^{'}_{prsq}=\left(g_s\left(c_{pq}+c_{qr}\right)\right)/\alpha$ and $U^{'}_{prs}=\left(g_s c_{pr}\right)/\alpha$ (13)

Pipeline inventory cost per unit product can be obtained from equation (8) and (9), and it is given by

$U^{"}_{prsq}=m_s(t_{pq}+t_{qr}+t_q)$ and $U^{"}_{prs}=m_s t_{pr}$ (14)

The objective function of the mathematical model is presented as:

Minimize = TTC + TIC + CEC + FCCD

Minimize

$$\left(\begin{array}{l}\sum_{p,r,s,q}\left(K_{prsq}U^{'}_{prsq}Y_{prsq}+K_{prs}U^{'}_{prs}Z_{prs}\right)+\sum_{p,r,s,q}\left(K_{prsq}U^{"}_{prsq}Y_{prsq}+K_{prs}U^{"}_{prs}Z_{prs}\right)+\sum_{p,r,q}\frac{\sum_{p,s}m_s\left(N_{pr}K_{prsq}+N_{qr}K_{prs}\right)}{2\left(N_{qr}*N_{pr}\right)}\\+\omega\delta\sum_{p,r,s,q}\left(d_{pq}+d_{qr}\right)\left(K_{prsq}v_sY_{prsq}\right)+\left(d_{pr}K_{prs}v_sZ_{prs}\right)+\sum_{q}X_qFC_q\end{array}\right)$$

(15)

The objective function given in equation (15) comprises five terms. The first term presents the total transportation cost, including the inbound, outbound and direct shipment. The second and third term represents the pipeline and retailer inventory cost, respectively. The fourth term shows the cost associated with the carbon emission emitted from the vehicles used for the shipment of the products through cross-dock or direct. The last term presents the fixed cost incurred for opening the cross-docks.

Subject to the following conditions.

$\sum_{r,q}K_{prsq}+\sum_{r}K_{prs}\leq A_{ps}$ $\forall\ p\in P,s\in S$ (16)

$\sum_{p,q}K_{prsq}+\sum_{p}K_{prs}=E_{rs}$ $\forall\ r\in R,s\in S$ (17)

$\sum_{p,r,s}K_{prsq}F_sY_{prsq}\leq\beta_qX_q$ $\forall\ q\in Q$ (18)

$K_{prs}\leq MZ_{prs}$ $\forall\ p\in P,r\in R,s\in S$ (19)

$K_{prsq},K_{prs}\geq 0$ $\forall\, p\in P,\ r\in R,\ s\in S,q\in Q$ (20)

$$N_{pq}, N_{qr}, N_{pr} \geq \square^{+} \qquad \forall\, p \in P, r \in R, q \in Q \tag{21}$$

$$X_{q}, Y_{prsq}, Z_{prs} \in \{0,1\} \qquad \forall\, p \in P,\ r \in R,\ s \in S,\ q \in Q \tag{22}$$

Equation (16) presents the supply constraint, which takes into consideration both direct shipment and indirect shipment via cross-docking facility. Equation (17) satisfies the demand of the retailers for different product types. Equation (18) states that the demand of the retailers should be met from the suppliers only via opened cross-docks and then the capacity restrictions for the cross-docking facility need to be maintained. Constraint (19) ensures that the direct shipment can be performed, if a supplier is assigned to a specific retailer using Big M constraint. The *M* is a sufficiently large number. Constraints set (20) - (22) present the non-negative integer variables, integer variables and binary variables, respectively.

4. Solution approach

The computation of the proposed mathematical model based on capacitated cross-docking problem becomes complicated, as the problem size increases exponentially with the increasing number of suppliers, cross-docks, retailers and product types; as the case company attempts to expand its business in the northern region of India. The computational effort and memory requirements for complex models are massive due to large number of variables and parameters, while employing the exact techniques (Maiyar and Thakkar 2019; Mogale et al., 2020). Metaheuristic algorithms play a crucial role in tackling the complicated and real-life high dimensional problems, which are challenging to solve using commercial solvers in a reasonable computational time (Chen et al., 2016; De et al., 2019b). Also, most conventional solvers are incapable of handling the non-linear equations of the mathematical model (Yu et al., 2017).

Further, the neighbourhood search of these algorithms becomes complicated to employ due to the concurrent integration of various decisions into the neighbourhood moves (Vidal et al., 2016; Tang and Wang, 2006). Neighbourhood search algorithms are commonly used for solving complex scheduling, location and vehicle routing problems (e.g., Şevkli and Güler, 2017; Sze et al., 2017). The motivation to use VNS and other metaheuristics were driven by their capability to solve complex mixed integer non-linear programming problems (e.g., Govindan et al., 2019; Rostami et al., 2020; Gruler et al., 2020). Thus, the VNS based algorithm is implemented, which is an effective metaheuristic and provides encouraging results for combinatorial problems in the literature (Kuo and Wang, 2012; Stenger et al., 2013). The VNS is relatively easy to implement due to the small number of parameters (Li and Tian, 2016; Govindan et al., 2019). The multiple integer non-linear programming model presented above is solved using a random search algorithm named two-level self-adaptive variable neighbourhood search (TLSAVNS) algorithm. TLSAVNS algorithm developed by Li and Tian (2016) is used to solve two-level decisions in the supply chain network. The solution obtained from the TLSAVNS is validated with the solution obtained through the VNS algorithm.

4.1. Variable Neighbourhood Search (VNS) algorithm

The VNS is a local search-based algorithm, which explores the solution space in the neighbourhood structures for escaping from the local entrapment (Hansen et al., 2010). A relation among these neighbourhood structures is established according to the problem undertaken (Alguwaizani et al., 2011). Components of variable neighbourhood search algorithm include an initial feasible solution, shaking procedure, first improvement, neighbourhood change and a terminating condition. Algorithm A.1 presents the pseudo-code for the VNS algorithm. Let x be an initial feasible solution and $f\left(x\right)$ be the objective function value considering the initial feasible solution. Let $N_k\left(x\right)$ be a set of solutions in the k^{th} neighbourhood structure where x is one of the solution. K_{max} is the maximum number of different neighbourhood structures generated in the shaking stage. In the shaking procedure, from the k^{th} neighbourhood structure, a solution x' is randomly chosen and updated in the following way, $x'' = x'$. Within First Improvement local search procedure, x'' is used as an initial solution and the local search operator searches the solutions $N_k\left(x\right)$ within the k^{th} neighbourhood structure. If the fitness function value of solution x'', $f(x'')$ is less than the fitness function value of x, $f\left(x\right)$, or $f(x'') < f\left(x\right)$, then the initial solution of the First Improvement local search procedure is updated as $x'' = x'$ and $f(x'') = f\left(x\right)$. The local search procedure also checks whether the fitness function value of the local best solution x'', $f(x'')$ is better than the fitness function value of the global best solution x^*, $f(x^*)$. If $f(x'') < f(x^*)$, then the global best solution is updated, $x^* = x''$ and the algorithm searches the next neighbourhood structure $N_{(k+1)}\left(x\right)$. For determining the fitness function values, the decision variable values given in each solution (please refer to figure 2 where each row of the neighbourhood structure depicts a solution comprising of decision variable values), is used and value of the objective function of the mathematical model (given in equation 15) is determined. Algorithm A.1 provides detailed procedure of the Variable Neighbourhood Search algorithm.

Within variable neighbourhood search algorithm, we have considered first improvement as the local search (for intensification) operator and shaking procedure as the perturbation operator (for diversification). First Improvement operator is employed as a single local search operator within VNS algorithm in past literature (Wassen et al., 2017, Samà et al., 2017, Todosijević et al., 2016). The first improvement procedure uses the initial solution x' and rigorously searches the given neighbourhood structure $N_k(x)$ (refer to figure 2 for an example of the encoding scheme of a neighbourhood structure) and compares each solution of the neighbourhood structure with the initial solution x'. If a better solution is obtained, then the overall best solution x" is updated. After performing the local search, the neighbourhood change takes place. Although, if an improvement is achieved in the first improvement procedure in terms of obtaining a better solution x", then the local search returns to its first neighbourhood structure (*k* = *1*) and updates the best-known solution. Otherwise, the algorithm keeps on performing the search on different neighbourhood structure (*k* = *k* + *1*) for obtaining a better solution. Once the local search is performed on all the neighbourhood structures, the VNS algorithm stores the best-known solution and moves on to the next iteration. In this way, the VNS performs its searching procedure and obtain the overall best solution with the objective function value $f\left(x^*\right)$. Furthermore, in order to resolve local optima traps where the proposed local search may be stuck, we consider the Shaking procedure as the perturbation operator. Shaking procedure randomly selects a solution x' from the k^{th} neighbourhood structure which is employed as an initial solution for the procedure. Past literature focusing on variable neighbourhood search algorithm have also considered first improvement as the local search operator and shaking procedure as the perturbation operator (Wassen et al., 2017, Samà et al., 2017, Todosijević et al., 2016). The VNS algorithm terminates after reaching the maximum number of iterations. Algorithm A.1 highlights the VNS algorithm with input parameters as *N* (Number of iterations), *Kmax* (Number of Neighbourhood structures), *x* (initial feasible solution) and *f(x)* (fitness function value of the initial feasible solution *x*). Output parameters of algorithms (1) are x^* (overall best solution obtained at the end of the number of iterations) and $f\left(x^*\right)$ (fitness function value for the overall best solution).

The parameters of the VNS algorithm are the number of iterations and the maximum number of the neighbourhood structures. The pseudo-code of the VNS algorithm is provided in algorithm A.1. Each neighbourhood structure considered in the VNS can be considered as a configuration of several suppliers, retailers, opened cross-docks and products to be transported, which results in the variables for units of goods (freight) and number of vehicles moving from one point to another. Each solution in the neighbourhood structure comprises of the values of the decision variables of the mathematical model – (i) Units of product *s* supplied by supplier *p* and shipped through cross-dock *q* to retailer *r* (K_{prsq}), (ii) Units of product *s* directly supplied by supplier *p* to retailer *r* (K_{prs}), (iii) Whether cross-dock *q* is open or not (X_q) and (iv) Whether supplier *p* is transferring the product *s* to retailer *r* through cross-dock *q* and (v) Whether supplier *p* is directly transferring the product *s* to retailer *r (Z_{prs}).* Moreover, the solutions also comprise of the number of trucks utilised for product shipment between suppliers to cross-docks (N_{pq}), suppliers to retailers (N_{qr}) and cross-docks to retailers (N_{pr}). Figure 2 presents the encoding scheme for Case 8 (3-30-1-2) where suppliers *P* = 3, retailers *R* = 30, products *S* = 1 and cross-docks *Q* = 2. Figure 2 comprises one neighbourhood structure which consists of 100 solutions presenting the values of 698 decision variables for the Case 8 (3-30-1-2).

Algorithm (A.1): The pseudo-code of Variable Neighbourhood Search (VNS)

```
Algorithm : Variable Neighbourhood Search
$(x^*, f(x^*)) = VNS(N,\ Kmax,\ x,\ f(x))$
$N$ = number of iterations
$Kmax$ = maximum number of neighbourhood structures
$N_k(x)$ = Set of solutions within $k^{th}$ neighbourhood structure
Here the output is $x^*$ with best fitness function as $f(x^*)$

Initialization
Initialize with a feasible solution $x$ of $f(x)$
Choose an initial feasible solution $x$ and compute $f(x)$
Set the current best local minimum
$x^* = x$ and $f(x^*) = f(x)$

Iterations
for $n = 1$ to $N$
    $k = 1$
    Generate $Kmax$ number of neighbourhood structures using Algorithms $C.1$ and $D.1$
    Following neighbourhood structures are obtained $(N_1(x), N_2(x), N_3(x), \ldots, N_{Kmax}(x))$
    while $k < Kmax$
        Shaking: obtain a random solution $x'$ belongs to $N_k(x) \rightarrow$ Update $x'' = x'$
        Perform Local search: FirstImprovement on $N_k(x)$ to update local minima $x''$
       if $f(x'') < f(x)$
            Set $k = 1$; $x'' = x$; $f(x'') = f(x)$
            if $f(x'') < f(x^*)$
                Set $x^* = x''$ ; $f(x^*) = f(x'')$
            end
       else
       Neighbourhood Change:
            Set $k = k + 1$
       end
    end
end
Return $x^*$ and $f(x^*)$
```

$$\begin{bmatrix}
1 & {}_{1}^{1}K_{prsq}^{180} & {}_{1}^{181}K_{prs}^{270} & {}_{1}^{271}X_{q}^{272} & {}_{1}^{273}Y_{prsq}^{452} & {}_{1}^{453}Z_{prs}^{542} & {}_{1}^{543}N_{pq}^{548} & {}_{1}^{549}N_{qr}^{608} & {}_{1}^{609}N_{pr}^{698} \\
2 & {}_{2}^{1}K_{prsq}^{180} & {}_{2}^{181}K_{prs}^{270} & {}_{2}^{271}X_{q}^{272} & {}_{2}^{273}Y_{prsq}^{452} & {}_{2}^{453}Z_{prs}^{542} & {}_{2}^{543}N_{pq}^{548} & {}_{2}^{549}N_{qr}^{608} & {}_{2}^{609}N_{pr}^{698} \\
. & & & & & & & & \\
. & & & & & & & & \\
. & & & & & & & & \\
100 & {}_{100}^{1}K_{prsq}^{180} & {}_{100}^{181}K_{prs}^{270} & {}_{100}^{271}X_{q}^{272} & {}_{100}^{273}Y_{prsq}^{452} & {}_{100}^{453}Z_{prs}^{542} & {}_{100}^{543}N_{pq}^{548} & {}_{100}^{549}N_{qr}^{608} & {}_{100}^{609}N_{pr}^{698}
\end{bmatrix}$$

Figure 2: Encoding scheme of a neighbourhood structure for the VNS algorithm which comprise of decision variables values for 100 solutions

4.2. Two-Level Self-Adaptive Variable Neighbourhood Search (TLSAVNS) algorithm

TLSAVNS algorithm is an improved variant of the VNS algorithm introduced by Li and Tian (2016). The TLSAVNS algorithm aims to adopt VNS at two levels for obtaining more promising results in less computational time. The TLSAVNS algorithm comprises of two parameters – maximum number of first level neighbourhood structure and maximum number of second level neighbourhood structure. Let us assume that the index of the first level neighbourhood structure as *m* and that of the second level neighbourhood structure as *n*. The first level neighbourhood structures are predetermined by random allocation and the second level neighbourhood structures are self-adaptively selected based on the previous searches rather than a pre-regulated sequence of the neighbourhood structures (Li and Tian, 2016). Two crucial features of this algorithm are, namely, two-level VNS architecture and self-adaptive technique of neighbourhood selection (more discussion about the self-adaptive technique is presented on section 4.3). Each neighbourhood structure in the second level is assigned with a selection ranking $sr_i > 0$, which is initially set to a constant value with a rough estimation.

In the first level of TLSAVNS, a solution is randomly chosen from the first neighbourhood structure using an important operator of the algorithm named as shaking function (N_m) which acts as a perturbation operator for diversification within the solution space (Hansen and Mladenović 2014). Now, First Improvement local search operator (for intensification) is performed on the current neighbourhood structure N_m. During the local search procedure for obtaining a local optimum on first level neighbourhood structure, this local optimum (say S_1) is compared with that of the second level neighbourhood structure (say S_2). If the second level search provides a better solution, then its selection ranking sr_i is updated to a better ranking (say by adding α). Li and Tian (2016) adopted the local search operator discussed over here as the intensification operator and Shaking procedure as perturbation operator. This procedure is repeated until the stopping condition is met (refer to Algorithm-2). The pseudo-code of the TLSAVNS algorithm is shown in algorithm B.1.

In algorithm B.1, the shaking procedure is adopted to obtain solutions a and b for first level and second level neighbourhood structures respectively. The shaking operator in the first level neighbourhood structure is referred to as shaking (a, N_a) and the corresponding local optimum a' is obtained using the local search operator. Within first level neighbourhood structure $N_k(a)$, First Improvement local search procedure is performed where a is used as an initial solution and the local search operator searches the solutions within the neighbourhood structure. If the fitness function value of any solution a" within the neighbourhood structure, $f(a")$ is less than the fitness function value of initial solution a, $f(a)$, or $f(a") < f(a)$, then the initial solution of the First Improvement local search procedure is updated as $a = a"$. Then, the local search procedure also checks whether the fitness function value of the updated initial solution a, $f(a)$ is better than the fitness function value of the local optimum a', $f(a')$. If $f(a) < f(a')$, then the local optimum value is updated, $a' = a$. For determining the fitness function, the values of decision variables mentioned in each solution is used to obtain the objective function of the mathematical model (given in equation 15).

In the second level, the shaking operator is denoted by shaking (b, N_b) and its local optimum b' is obtained using local search operator. Within second level neighbourhood structure $N_k(b)$, First Improvement local search procedure is adopted considering b as the initial solution. If the fitness function value of any solution $b"$, $f(b")$ within the second level neighbourhood structure is less than the fitness function value of the initial solution b, $f(b)$ or $f(b") < f(b)$, then the initial solution is updated $b = b"$. Furthermore, the algorithm compares the fitness function value of the initial solution with the fitness value of the local optimum solution, if $f(b) < f(b')$, the local optimum solution is updated in the following way, $b' = b$.

Now, if the optimum solution b' of second level neighbourhood structure $N_k\left(b\right)$, is better than local optimum a' of the first level neighbourhood structure $N_k\left(a\right)$, so if $f(b') < f(a')$ then the selection probability is upgraded. Else, it is downgraded, and the searching procedure moves to the next neighbourhood structure $N_{\left(k+1\right)}\left(b\right)$ within the second level neighbourhood structure. Section 4.3 highlights the detailed procedure of the neighbourhood selection and update of selection probability. The main advantage of this algorithm is that the adaptive technique for neighbourhood selection allows only neighbourhood structure with a better solution, owing to selection probability (Li and Tian, 2016). This approach not only makes the search more competent, but also enhances the robustness of the basic VNS. The L used in the pseudo-code of the TLSAVNS algorithm represents a maximum number of iterations.

4.3. Neighbourhood selection and update of selection probability

In the TLSAVNS algorithm, the first level neighbourhood follows the conventional VNS procedure having the pre-determined sequence of neighbourhood structures, whereas the second level neighbourhood of the TLSAVNS algorithm is self-adaptively selected based on its selection probability instead of the pre-determined sequence. The selection probability of each neighbourhood structure on the second level neighbourhood is dependent on the concept of success and failure of a neighbourhood search. If the local search procedure on a neighbourhood structure results in a new solution and if it's better than the input solution, then the neighbourhood structure is viewed as a successful one. When no better solution is obtained from the neighbourhood structure, then the neighbourhood structure is viewed as unsuccessful. The selection probability operator is initially set during the first few tests runs so that the algorithm is trained, and it can analyse the suitability and performance of the neighbourhood structures. The algorithm B.1 calculates the success count as s_i and failure count as f_i for each of the neighbourhood structure. The selection probability of each neighbourhood N'_k is computed using the following equation:

$$SP_k = S_k \Big/ \sum_{i=1}^{K_2} S_i \,, \tag{23}$$

Here, S_i is the success ratio, and it is computed using the following equation

$$S_i = \left[s_i / \left(s_i + f_i \right) \right] + 0.05 \,,\; i = 1,...,K_2 \tag{24}$$

The value 0.05 is added to each S_i in order to ensure that the selection probability of some neighbourhoods can avoid the value zero if better solutions are not obtained in the previous search. Based on the computation method presented in equations (23) and (24), the TLSAVNS algorithm ensures that successful neighbourhood structures with larger selection probability are employed to generate new solutions. The self-adaptive mechanism helps the TLSAVNS to reduce the burden of computational complexity, as the local search is performed on the successful neighbourhood structures only. Pseudo-code of the TLSAVNS Algorithm is presented in Algorithm B.1 to highlight the ways operators of the TLSAVNS such as shaking operator, local search operator and selection probability operator interacts with each other. Input parameters of Algorithm B.1 are *L* (Number of iterations), N_k (set of Neighbourhood structure for first level), N'_k (set of Neighbourhood structure for the second level), *x* (initial feasible solution) and *f(x)* (functional value or fitness function value of the initial feasible solution *x*). Output parameters of the Algorithm B.1 comprises of *x* (the overall best solution at the end of the iteration) and *f(x)* (functional value or fitness function value of the overall best solution *x*).

Algorithm (B.1): The pseudo-code of Two-Level Self-Adaptive Variable Neighbourhood Search

```
Algorithm : $Two-Level\ Self-Adaptive\ Variable\ Neighbourhood\ Search$
$(x, f(x)) = TLSAVNS(L, N_k, N'_k, x, f(x))$
Initialize a set the neighborhood structure of the first level $N_k\ (k = 1,\ldots,K_1)$ and the
            second level $N'_k\ (k' = 1,\ldots,K_2)$ using Algorithms $C.1$ and $D.1$
Initialize the selection probability of each neighborhood $N'_k$ to be $1/K_2$, and $l = 0$
Repeat until stopping condition is reached $(l < L)$
      While( $k \le K_1$ )
         Generate a random new solution in neighborhood $N_k = a$
         Perform local search using value a in neighborhood $N_k = a'$
         $t = 0$
         While $(t < T)$
            Generate a random new solution in neighborhood $N'_k = b$
            Perform local search using value b in neighborhood $N'_k = b'$
            if functional value$(b')$ < functional value$(a')$
               $a' = b'$
               Update Probability$(N'_k)$
            else
               Update Probability$(N'_k)$
               $ns =$ Neighborhood Selection$(N'_k)$
            End if
            $t = t + 1$
         End while
         if functional value$(a')$ < functional value$(x)$
            $x = a'$
            Return to the first neighborhood of $N_k$
         else
            Turn to the next neighborhood of $N_k$
         End if
         $k = k + 1$
      End while
      $l = l + 1$
End
Return x and $f(x)$
```

4.4. Initial Solution Generation Procedure – Building Neighbourhood Structure

The sub-section aims to highlight the detailed procedure adopted to obtain the initial solution in the form of a neighbourhood structure associated, which comprises of decision variables of the mathematical model. The neighbourhood structure is developed considering the problem structure and thus, helps to eliminate the infeasible solutions which arise during random solution generation (De et al. 2020).

Several capacitated vehicles available at supplier are used to transfer the products directly to retailers or through cross-docks. Initially, suppliers select the capacitated vehicle following the demand of the retailer and fixed costs of the vehicle. If the demand of a particular retailer is sufficient for full truckload capacity, then the supplier satisfies the demand of that retailer through direct transfer. Using equation (25), the value of λ_{rs} is determined where λ_{rs} is a positive integer value, determining whether supplier decides to use full truckload capacity and satisfies the demand of the retailer through direct shipment.

$$\lambda_{rs} = \left[\frac{E_{rs}}{\alpha}\right] \qquad \forall\ r \in R, s \in S \qquad (25)$$

When λ_{rs} is an positive integer value, then a supplier nearest to the particular retailer in terms of distances is chosen for employing the direct shipment for satisfying the demand of the retailer. Therefore, considering the distances d_{pr} between the particular retailer and for all the suppliers. Now, identifying the supplier p_{min} for which the $d_{p_{min}r}$ is minimum for the particular retailer *r* while considering all the suppliers. As the direct shipment is adopted in this case, hence the value of the decision variables $K_{p_{min}rs}$ and $Z_{p_{min}rs}$ are determined (here, $p = p_{min}$) using equations (26) and (27) given below,

$$K_{p_{min}rs} = E_{rs},\ \ For\ p = p_{min}\ and\ \lambda_{rs} = positive\ integer \qquad \forall\ r \in R, s \in S \qquad (26)$$

$$Z_{p_{min}rs} = \begin{cases} 1, & For\ \lambda_{rs} = positive\ integer \\ 0, & For\ \lambda_{rs} = fraction\ value \end{cases} \qquad For\ p = p_{min} \qquad \forall\ r \in R, s \in S \qquad (27)$$

The value of the decision variable K_{prs} helps to determine the decision variable N_{pr}, related to the number of trucks utilised for the shipment of products from the supplier to the retailer using equation (5). Algorithm C.1 presents a detailed procedure for the generation of the decision variables K_{prs}, Z_{prs} and Y_{prsq}.

When the value of λ_{rs} is in fraction, the supplier decides to use the cross-docking facility for meeting the demand of the respective retailer. In such a case, the distance of all the cross-docks from the particular retailer *r* is determined and accordingly the cross-dock q_{min}, with least distance

$d_{q_{min}r}$ from the specific retailer *r* is selected. Now, the distance of all the suppliers from the selected cross-dock q_{min} is taken into consideration and accordingly the supplier p_{min} with least distance $d_{p_{min}q_{min}}$ from the cross-dock q_{min} is preferred. Therefore, the value of decision variable $Y_{p_{min}rsq_{min}}$ is obtained for supplier $p = p_{min}$ and cross-dock $q = q_{min}$ and accordingly presented in equation (28).

$$Y_{p_{min}rsq_{min}} = \begin{cases} 1, & For\ \lambda_{rs} = fraction\ value \\ 0, & For\ \lambda_{rs} = positive\ integer \end{cases} \quad For\ p = p_{min},\ q = q_{min} \qquad \forall\ r \in R, s \in S \qquad (28)$$

Suppliers select the route with minimum distance for transferring the products to the cross-docking facility and finally to the retailers. Therefore, equation (25) helps supplier in deciding whether to meet the demand of a certain product type for a specific retailer through direct shipment or via cross-dock facility. If the demand is not sufficient to fulfil the full truck capacity, then the supplier combines the demand of multiple retailers to fully load the truck and transport it through the cross-docking facility. Algorithm C.1 computes the value of a binary variable Y_{prsq} which gives necessary information regarding the transportation route to be accessed from supplier to the retailer via specific cross-dock. Input parameters of Algorithm C.1 are number of product types *S*, number of retailers *R*, number of suppliers *P*, number of cross-docks *Q*, demand of retailer for the product type E_{rs}, capacity of truck α and distance parameters d_{pr}, d_{qr}, d_{pq}. Output parameters of Algorithm C.1 are decision variables K_{prs}, Z_{prs} and Y_{prsq}.

Depending upon the available capacity of the preferred supplier for a certain product type, the cross-dock determines the amount of product to be shipped from the supplier to the cross-dock. If the capacity of the product type available with the supplier is more than the demand of the retailer for the product type, then the demand of the retailer is met from the supplier and accordingly the decision variable K_{prsq}, is updated. Although, when the capacity available with the supplier for the specific product type is less than the demand of the retailer for the product type, then the available supplier capacity is sent to the retailer and the value of the decision variable K_{prsq} is updated. Equation (29) aims to determine the value of the decision variables K_{prsq}, when

$p = p_{min}$, $q = q_{min}$ for a specific retailer and $Y_{p_{min} r s q_{min}} = 1$. The value of the K_{prsq} is used to estimate the value of the decision variables N_{pq} and N_{qr} using equations (1) and (3) respectively. The variables N_{pq} and N_{qr} are related to the number of trucks utilised for product shipment from suppliers to cross-docks and number of trucks deployed from cross-docks to retailers respectively.

$$K_{p_{min} r s q_{min}} = \begin{cases} E_{rs}, \ A_{ps} \geq E_{rs} \\ A_{ps}, \ A_{ps} < E_{rs} \end{cases} \quad For\ Y_{p_{min} r s q_{min}} = 1 \ \& \ p = p_{min}, \ q = q_{min} \quad \forall \ r \in R, s \in S \qquad (29)$$

When the capacity of the supplier for the product type is less than the demand of the retailer or $A_{ps} < E_{rs}$, then the remaining amount $\left(E_{rs} - A_{ps}\right)$ pertaining to the demand of the retailer is met via different strategy, where the cross-docking facility randomly chooses another supplier p^{rand}. The cross-dock checks whether the chosen supplier p^{rand} has enough capacity or not to meet the remaining demand $\left(E_{rs} - A_{ps}\right)$ of the specific retailer. The algorithm D.1 gives the detailed procedure for the generation of the variables K_{prs} and Y_{prsq}. Based on the value obtained for binary variable Y_{prsq}, the value of binary variable X_q is updated using equation (30).

$$X_q = \begin{cases} 1, & Y_{prsq} = 1 \\ 0, & Y_{prsq} = 0 \end{cases} \qquad \forall \ p \in P, r \in R, s \in S, q \in Q \qquad (30)$$

Input parameters of Algorithm D.1 are number of product types *S*, number of retailers *R*, number of suppliers *P*, number of cross-docks *Q*, demand of retailer for the product type E_{rs}, decision variable Y_{prsq} and capacity of supplier for the product type A_{ps}. Output parameters of Algorithm D.1 are decision variables K_{prs} and Y_{prsq}. It must be noted that the algorithm randomly selects the supplier for meeting the remaining demand and this random selection helps to obtain different feasible solutions, which are fed into the VNS algorithm and TLSAVNS algorithm.

Algorithms C.1 helps in obtaining the values of the decision variables K_{prs} (number of product *s* directly supplied by supplier *p* to retailer *r*), Z_{prs} (whether supplier *p* is directly transferring the product *s* to retailer *r*), Y_{prsq} (whether supplier *p* transfers product *s* to retailer *r* via cross dock *q*). Algorithm D1 assists in determining the values of decision variables K_{prsq} (number of product *s* supplied by supplier *p* and shipped through cross-dock *q* to retailer *r*) and updating the values of Y_{prsq}. Using the values of the decision variables from algorithms C.1 and D.1, we obtain the values of the number of trucks utilised for product shipment between suppliers to cross-docks (N_{pq}), suppliers to retailers (N_{qr}) and cross-docks to retailers (N_{pr}). Therefore, algorithms C.1 and D.1 determines values of the decision variables which comprise of one solution within the neighbourhood structure given in figure 2. Accordingly, algorithms C.1 and D.1 are employed to obtained the values for all the solutions with the neighbourhood structure given in figure 2. Variable neighbourhood search algorithm (pseudo code given in Algorithm A.1) uses algorithms C.1 and D.1 to determine K_{max} number of different neighbourhood structures, used in every iteration within the algorithm A.1. Furthermore, TLSAVNS algorithm (pseudo-code given in algorithm B.1) uses algorithms C.1 and D.1 to determine K_1 and K_2 associated with number of different neighbourhood structures for first and second levels respectively.

Algorithm C.1: Procedure about the generation of the variables K_{prs}, Z_{prs} and Y_{prsq}

```
$Procedure: Generation\ of\ decision\ variables\ K_{prs}, Z_{prs}\ and\ Y_{prsq}$
$\left(K_{prs}, Z_{prs}\ and\ Y_{prsq}\right) = Algorithm\_3\left(S,\ R,\ P,\ Q,\ E_{rs},\ \alpha,\ d_{pr},\ d_{qr},\ d_{pq}\right)$
 $for\ s = 1\ to\ number\ of\ product\ types,\ S$
    $for\ r = 1\ to\ number\ of\ retailers,\ R$
        $Determine\ \lambda_{rs} = \left[E_{rs}/\alpha\right]$
        $if\ \lambda_{rs} = positive\ integer$
            $for\ p = 1\ to\ number\ of\ suppliers,\ P$
                $Obtain\ d_{pr}$
            $end$
            $Identify\ p_{min},\ for\ which\ d_{p_{min}r}\ is\ minimum\ among\ all\ suppliers$
            $for\ p = 1\ to\ number\ of\ suppliers,\ P$
               $if\ p = p_{min}$
                   $Assign\ the\ value\ of\ K_{prs}\ as\ E_{rs},\ or\ K_{p_{min}rs} = E_{rs},\ where\ p = p_{min}$
                   $Assign\ the\ value\ of\ Z_{prs}\ as\ 1,\ or\ Z_{p_{min}rs} = 1,\ where\ p = p_{min}$
               $end\ if$
               $Assign\ the\ value\ of\ K_{prs}\ as\ 0,\ or\ K_{prs} = 0,\ where\ p \neq p_{min}$
               $Assign\ the\ value\ of\ Z_{prs}\ as\ 0,\ or\ Z_{prs} = 0,\ where\ p \neq p_{min}$
            $end$
        $elseif\ \lambda_{rs} = fraction\ value$
            $for\ q = 1\ to\ number\ of\ cross-docks,\ Q$
               $Obtain\ d_{qr}$
            $end$
            $Identify\ q_{min},\ for\ which\ d_{q_{min}r}\ is\ minimum\ among\ all\ cross-docks$
            $for\ p = 1\ to\ number\ of\ suppliers,\ P$
                $Obtain\ d_{pq_{min}}$
            $end$
            $Identify\ p_{min},\ for\ which\ d_{p_{min}q_{min}}\ is\ minimum\ among\ all\ suppliers$
            $Assign\ the\ value\ of\ Y_{p_{min}rsq_{min}}\ as\ 1,\ or\ Y_{p_{min}rsq_{min}} = 1,\ where\ p = p_{min},\ q = q_{min}$
        $end\ if$
    $end\ for$
 $end\ for$
$Return\ K_{prs}, Z_{prs}\ and\ Y_{prsq}$
```

Algorithm D.1: Procedure about the generation of the variables K_{prsq} and Y_{prsq}

```
$Procedure: Generation\ of\ decision\ variables\ K_{prsq}\ and\ Y_{prsq}$
$(K_{prsq}, Y_{prsq}) = Algorithm\_4(S, R, Q, P, Y_{prsq}, A_{ps}, E_{rs})$
$for\ s = 1\ to\ number\ of\ product\ types,\ r = 1\ to\ number\ of\ retailers,$
    $q = 1\ to\ number\ of\ cross-docking\ facilities,\ p = 1\ to\ number\ of\ suppliers$
    $if\ value\ of\ Y_{prsq}\ is\ 1,\ or\ Y_{prsq} = 1$
        $Obtain\ E_{rs},\ or\ the\ demand\ of\ product\ type\ s\ at\ retailer\ r$
        $Check\ the\ capacity\ available\ with\ supplier\ p\ for\ product\ type\ s,\ A_{ps}$
        $if\ available\ capacity\ with\ supplier\ p,\ A_{ps} \geq E_{rs}$
            $Assign\ the\ value\ of\ K_{prsq}\ as\ E_{rs},\ or\ K_{prsq} = E_{rs}$
            $Update\ A_{ps} = A_{ps} - K_{prsq}$
        $elseif\ available\ capacity\ with\ supplier\ p,\ A_{ps} < E_{rs}$
            $Assign\ the\ value\ of\ K_{prsq}\ as\ A_{ps},\ or\ K_{prsq} = A_{ps}$
            $Update\ A_{ps} = 0,\ and\ Remaining\ Amount\ to\ be\ met,\ Cp\ is\ (E_{rs} - A_{ps})$
            $While\ Cp = (E_{rs} - A_{ps})$
                $Random\ Selection\ Step: Randomly\ select\ a\ supplier\ and\ term\ it\ as\ p^{rand}$
                $if\ available\ capacity\ with\ supplier\ p^{rand},\ A_{p^{rand}s} \geq (E_{rs} - A_{ps})$
                    $Assign\ the\ value\ of\ K_{p^{rand}rsq} = (E_{rs} - A_{ps}),\ Y_{p^{rand}rsq} = 1$
                    $Update\ A_{p^{rand}s} = A_{p^{rand}s} - K_{p^{rand}rsq},\ and\ Cp = 0$
                $elseif\ available\ capacity\ with\ supplier\ p^{rand},\ A_{p^{rand}s} < (E_{rs} - A_{ps})$
                    $Go\ to\ Random\ Selection\ Step\ to\ select\ another\ supplier$
                $end\ if$
            $end\ While$
        $end\ if$
    $end\ if$
$end\ for$
$Return\ K_{prsq}\ and\ Y_{prsq}$
```

5. Results and discussions

Different scenarios of a three-echelon supply chain network are considered by varying the number of suppliers, cross-docks, retailers, and product types. Table 2 presents the complexities associated with each problem case in terms of the number of integer variables, binary variables and constraints involved for a combination of suppliers (P), retailers (R), cross-docks (Q) and number of product types (S). All the cases are generated following the structure of retail distribution company to replicate a typical supply chain (SC) network. Since the case organisation does not have an established SC network in the northern region, to analyse the developed model, the computational experiments conducted used reliable (mainly secondary) source data as presented in Table 3 (ranges for parameters).

Three board case scenarios considered are based on the total number of variables involved in the model. Case numbers 1 to 7 belong to the group of problem sizes with less than 150 variables and termed *small-sized* cases. Case numbers 8 to 17 belong to the group of problem sizes with less than 1000 variables and termed *medium-sized* cases. Case numbers 18 to 24 belong to the group of problem sizes with greater than 1000 variables and termed *large-sized* cases. Both the algorithms- VNS and TLSAVNS are applied to all three case scenarios.

5.1 Parameter settings

Parameter setting was performed, as it aimed to achieve a near-optimal solution in the least computational time. In the beginning, the parameters of VNS algorithm (such as number of iterations, maximum number of neighbourhood structures (*Kmax*) and number of solutions within a neighbourhood structure) and parameters of TLSAVNS (such as number of iterations, maximum number of neighbourhood structures for two levels and number of solutions within each neighbourhood structure) were assigned values at random, and then the algorithm was run changing one parameter at a time, keeping other parameters constant until the best value of the parameter was obtained. A similar process for obtaining the values of parameters was followed in several published works such as Li et al. (2016), Mogale et al. (2020) and De et al. (2019b) and such procedure is acceptable in this research domain. Following the procedure, it was found that the decisive number of solutions within a neighbourhood structure of the VNS algorithm is 100. In the case of TLSAVNS, the maximum number of neighbourhood structures for the first level neighbourhood was found to be 10 and that of the second level neighbourhood structures to be 6, computed over 100 instances. Maximum number of neighbourhood structures, *Kmax* considered for VNS algorithm was found to be 20. The algorithm was iterated 200 times for the VNS and TLSAVNS approach. The computational experiment was conducted on a large data set after appropriately tuning the algorithmic parameters. All the experimental scenarios provided in Table 2 were solved on MATLAB R2018a software having 8 GB RAM with Intel Core i7 1.8GHz processor and 64-bit Operating System of Windows 8. The performance of both the algorithms was compared with each other to observe the differences in results and to develop useful insights.

Table 2. Supply chain network model complexity for varying problem sizes

Problem size	Case type	P (Suppliers)	R (Retailers)	S (Products)	Q (Cross-docks)	Variables			Constraints
						Integer	Binary	Total	
Small	Case 1 (2-2-1-2)	2	2	1	2	24	14	38	34
	Case 2 (2-3-1-3)	2	3	1	3	45	27	72	68
	Case 3 (3-10-1-2)	3	10	1	2	146	92	238	225
	Case 4 (2-3-1-10)	2	3	1	10	122	76	198	201
	Case 5 (10-3-1-3)	10	3	1	3	189	123	312	376
	Case 6 (3-20-1-2)	3	20	1	2	286	182	468	445
	Case 7 (2-3-1-20)	2	3	1	20	232	146	378	391
Medium	Case 8 (3-30-1-2)	3	30	1	2	426	272	698	665
	Case 9 (20-3-1-3)	20	3	1	3	369	243	612	626
	Case 10 (2-3-1-30)	2	3	1	30	342	216	558	581
	Case 11 (30-3-1-3)	30	3	1	3	549	363	912	936
	Case 12 (3-50-1-2)	3	50	1	2	706	452	1158	1105
	Case 13 (2-3-1-50)	2	3	1	50	562	356	918	961
	Case 14 (50-3-1-3)	50	3	1	3	909	603	1512	1556
	Case 15 (3-100-1-2)	3	100	1	2	1406	902	2308	2205
	Case 16 (2-3-1-100)	2	3	1	100	1112	706	1818	1911
	Case 17 (100-3-1-3)	100	3	1	3	1809	1203	3012	3106
Large	Case 18 (12-14-9-13)	12	14	9	13	21674	21181	42855	43255
	Case 19 (27-17-3-17)	27	17	3	17	25993	24803	50796	56147
	Case 20 (27-10-4-26)	27	10	4	26	30392	29186	59578	64434
	Case 21 (24-17-6-16)	24	17	6	16	42680	41632	84312	87574
	Case 22 (16-26-5-22)	16	26	5	22	49810	47862	97042	102984
	Case 23 (27-8-7-42)	27	8	7	42	66702	65058	131760	137879
	Case 24 (28-15-8-33)	28	15	8	33	116079	114273	230352	239357
	Case 25 (29-23-9-30)	29	23	9	30	188320	186123	374443	386691

Table 3. Data span of parameters related to the model

Parameter	**Range of value**	**Parameter**	**Range of value**
c_{pq}	*[1,10000]*	t_q	*[.1,.2]*
t_{pq}	*[0,2]*	FC_q	*[10000-20000]*
d_{pq}	*[1,1000]*	E_{rs}	*[5,15]*
c_{qr}	*[1,10000]*	g_s	*[1,10]*
t_{qr}	*[0,2]*	m_s	*[1,10]*
d_{qr}	*[1,1000]*	v_s	*[1,15]*
c_{pr}	*[1,15000]*	F_s	*[.1,.2]*

t_{pr}	*[0,5]*	ω	*.0013*
d_{pr}	*[1,1000]*	δ	*[2,7]*
α	[100,1000]	β	*1.5*Total Product Demand*

5.2 The effectiveness of the algorithms

In this section, the results obtained by solving the mixed-integer non-linear programming model using VNS and TLSAVNS algorithms are tabulated. Table 4 presents the total cost incurred, considering both the algorithms along with the percentage variation in mean total cost with respect to the minimum total cost obtained. It is evident from Figure 3 that the TLSAVNS algorithm converges in a smaller number of iterations, as compared to the VNS algorithm. The TLSAVNS algorithm also solves the problem instances in less computational time compared to VNS and provides a solution nearer to the central tendency, even for large size problems (Table 4). Furthermore, to evaluate the performance of the algorithms, a graphical approach is applied as shown in Figure 4 on the small-sized, medium-sized and large-sized case scenarios. It is evident from these results that TLSAVNS is a more efficient algorithm for obtaining a near-optimal solution.

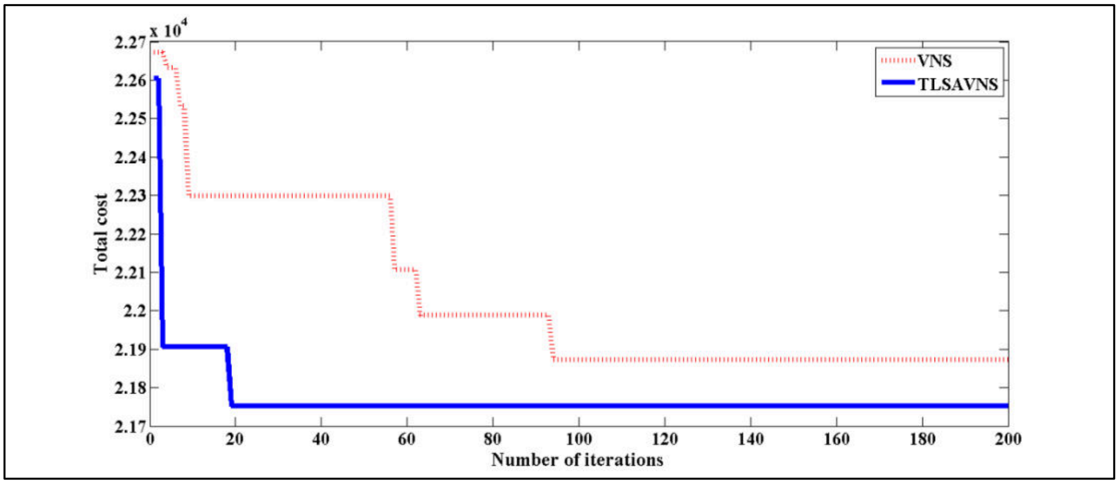

Figure 3. Convergence graph for Case 1 (2-2-1-2)

Figure 3 Alt text: Convergence graph for VNS and TLSAVNS showing the number of iterations on X-axis and total cost on Y-axis

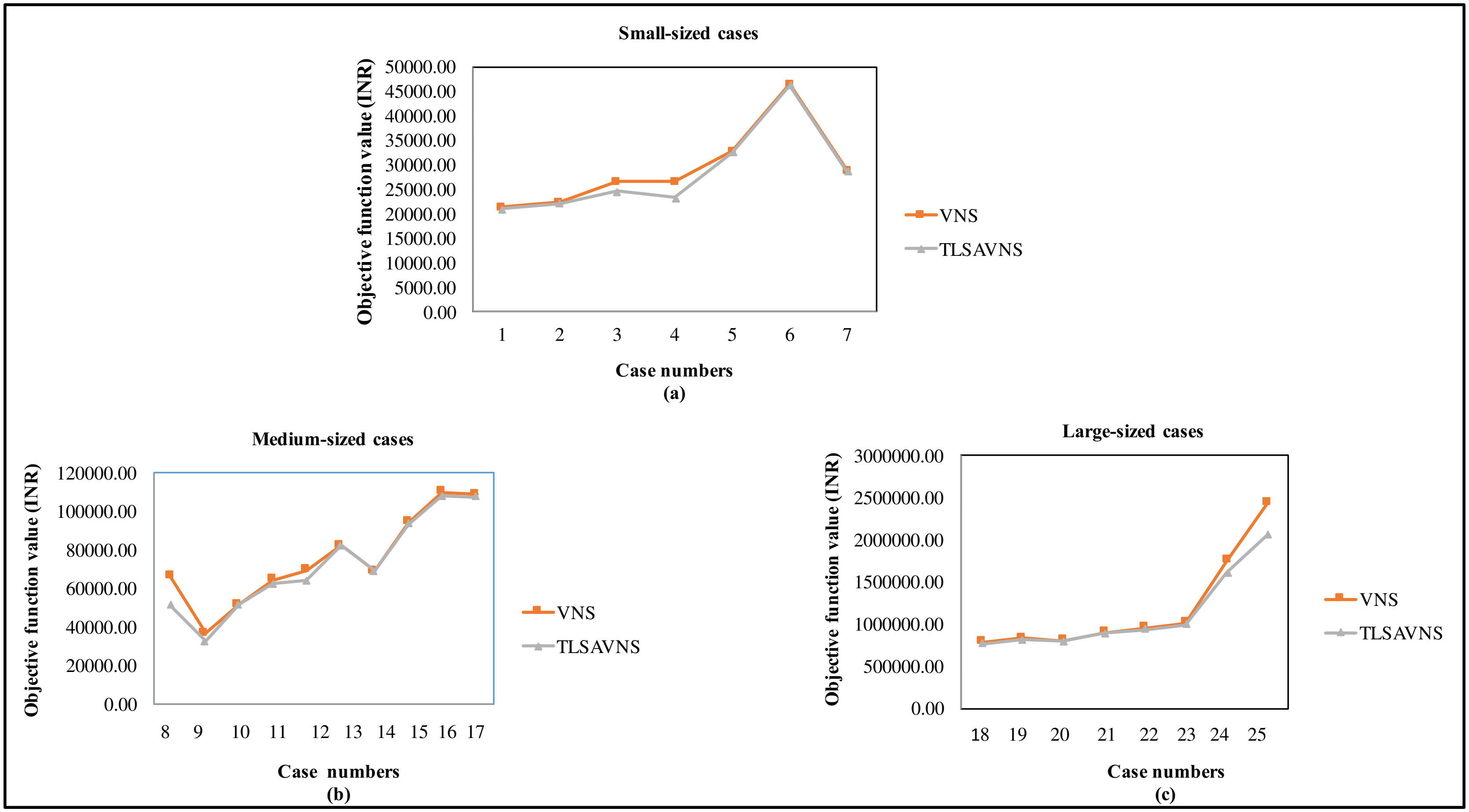

Figure 4. VNS and TLSAVNS objective function performance in a) Small-sized problems b) Medium-sized problems c) Large-sized problems. Line chart showing the performance of two algorithms in terms of objective functions for selected cases

Table 4. Computational results obtained from VNS and TLSAVNS for all the problem sizes

Problem Size	Case Type	Computational time (s)		Total Cost (INR)		% Variations in Cost		Carbon emission Cost		% Carbon emission variation	
		VNS	TLSAVNS	VNS	TLSAVNS	VNS	TLSAVNS	VNS	TLSAVNS	VNS	TLSAVNS
Small-sized	Case 1 (2-2-1-2)	42.7	28.7	21365.53	21033.74	0.16	0.59	18.56	17.95	0.07	0.09
	Case 2 (2-3-2-1-3)	83.5	60.5	22378.69	22178.5	0.89	1.12	980.97	960.35	4.38	4.33
	Case 3 (3-10-1-2)	311.2	208.6	26617.84	24674.89	0.51	0.76	892.55	830.46	3.35	3.37
	Case 4 (2-3-1-10)	223.5	174.6	26479.74	23215.45	0.84	0.55	540.79	514	2.04	2.21
	Case 5 (10-3-1-3)	371.2	293.5	32760.72	32531.58	0.3	1.03	1370.24	1299.94	3.57	3.34
	Case 6 (3-20-1-2)	454.7	328.1	46462.68	46301.95	1.26	1.54	1551.38	1668.17	3.41	3.59
	Case 7 (2-3-1-20)	368.5	332.5	28821.3	28714.19	0.3	1.02	573.18	535.07	2.02	1.86
Medium-sized	Case 8 (3-30-1-2)	656.8	497	65944.17	51063.76	1.25	0.67	793.87	790.87	1.17	1.55
	Case 9 (20-3-1-3)	624.7	525.8	36638.18	32270.91	0.51	1.24	383.36	347.85	1.05	1.08
	Case 10 (2-3-1-30)	536	461.3	51625.15	51614.04	0.77	0.88	1781.09	1701.54	3.45	1.36
	Case 11 (30-3-1-3)	885	720.6	64253.23	62192.09	1.97	0.11	1771.81	1724.7	2.45	2.77
	Case 12 (3-50-1-2)	636.4	494.4	69169.18	63753.72	0.41	1.63	778.09	776.43	1.11	1.22
	Case 13 (2-3-1-50)	917.8	690.9	82338.33	82260.12	0.83	0.28	4987.78	4937.8	5.91	4.79
	Case 14 (50-3-1-3)	1062.9	843.3	69074.21	69039.9	1.11	0.68	2256.66	2239.61	3.07	3.24
	Case 15 (3-100-1-2)	1035.3	991.3	93997.43	93451.4	1.05	0.38	1678.86	1525.88	1.79	1.63
	Case 16 (2-3-1-100)	1442.2	1094.5	109758.71	107865.84	1.52	0.62	1488.87	1486.53	1.35	1.38
	Case 17 (100-3-1-3)	1558.3	1152.5	108594.9	107277.1	0.93	1.02	3338.94	3114.27	2.57	2.9
Large-sized	Case 18 (12-14-9-13)	3324.1	2209.58	786334.5	763610.4	5.27	1.37	10565.32	10443.97	1.29	1.37
	Case 19 (27-17-3-17)	3418.3	2961.45	827399.51	808338.25	2.02	2.11	17227.8	17067.53	2.08	2.11
	Case 20 (27-10-4-26)	3283.8	2091.36	798865.2	795069.32	7.78	3.28	26060.18	26051.4	3.14	3.28
	Case 21 (24-17-6-16)	4639.8	3169.95	892350.89	886133.92	7.19	2.22	19654.6	19647.14	2.25	2.22
	Case 22 (16-26-5-22)	4657.4	3751.08	953787.1	932568.3	4.67	3.36	31329.52	31292.97	3.18	3.36
	Case 23 (27-8-7-42)	3403.4	2290.7	1014346.9	991562.2	7.34	4.34	43360.13	43035.43	4.27	4.34
	Case 24 (28-15-8-33)	6443.9	5084.28	1754262.5	1607069.4	8.11	5.97	95899.18	95895.21	5.3	5.97
	Case 25 (29-23-9-30)	7235.3	6451.1	2438759.08	2066201.2	9.21	4.28	172275.8	171164.61	6.57	8.28

5.3 Analysis of Results

The solution presented in this section showcases the relationship between prominent factors of the problem. Figure 5 explains the structure of the supply chain network, enlisting the various input parameters of case 1. The transportation links specify that the products are transported through the shown path. The values of the inbound transportation cost for that path (c), distance between supplier *p* and cross-dock *q* (d_{pq}), distance between cross-dock *q* and retailer *r* (d_{qr}) and the units of product moving in that path (K) are shown on the transportation link. The capacity of cross-dock *q*, and the demand at retailer *r* is mentioned in figure 5. It is evident that the number of units moved is more for the routes with less transportation cost and minimum distance.

It is to be noted that, Table 4 also quantifies the effect of the imposition of carbon emission cost on a supply chain network. It illustrates the amount of carbon emission cost incurred and the percentage of variation of the carbon emission cost with respect to the overall supply chain cost. The carbon tax (cost) is imposed for polluting the environment, and this cost is expected to be higher for the complete supply chain network, thus compelling that the concerned authorities need to rethink their freight transportation network and aim to design a more sustainable supply chain network. A comparison between the total number of available cross-docks and the total number of open cross-docks under different demand conditions is presented in Table 5. Here, four cases are evaluated as an example under varying demand conditions. The table signifies an increase in the number of open cross-docks with an increase in demand. The study explored the changes in the capacity of the cross-docks in relation to the number of open cross docks for a deterministic demand. The capacity of cross-dock *q* is varied from $0.5\times E_r \; to \; 5\times E_r$, where, $E_r = \left[Max\left(\sum_s E_{rs} \right) \; \forall r \right]$. It was observed that as the capacity of the cross-docks increases, the number of open cross-docks decreases for a given demand. This result can be used to analyse whether it is profitable to establish a single cross-dock with higher capacity or multiple cross-docks with smaller capacity.

Furthermore, the study provides insights into the number of suppliers and retailers that are affected due to the opening of cross-docks. To study this, the number of suppliers and retailers were varied keeping other factors constant. It is found that the number of suppliers does not affect the number of open cross-docks until the number of suppliers is slightly high. It is also observed that the number of open cross-docks increases with an increase in the number of retailers. An increase in retailers over suppliers, significantly influences the number of open cross-docks is another useful inference.

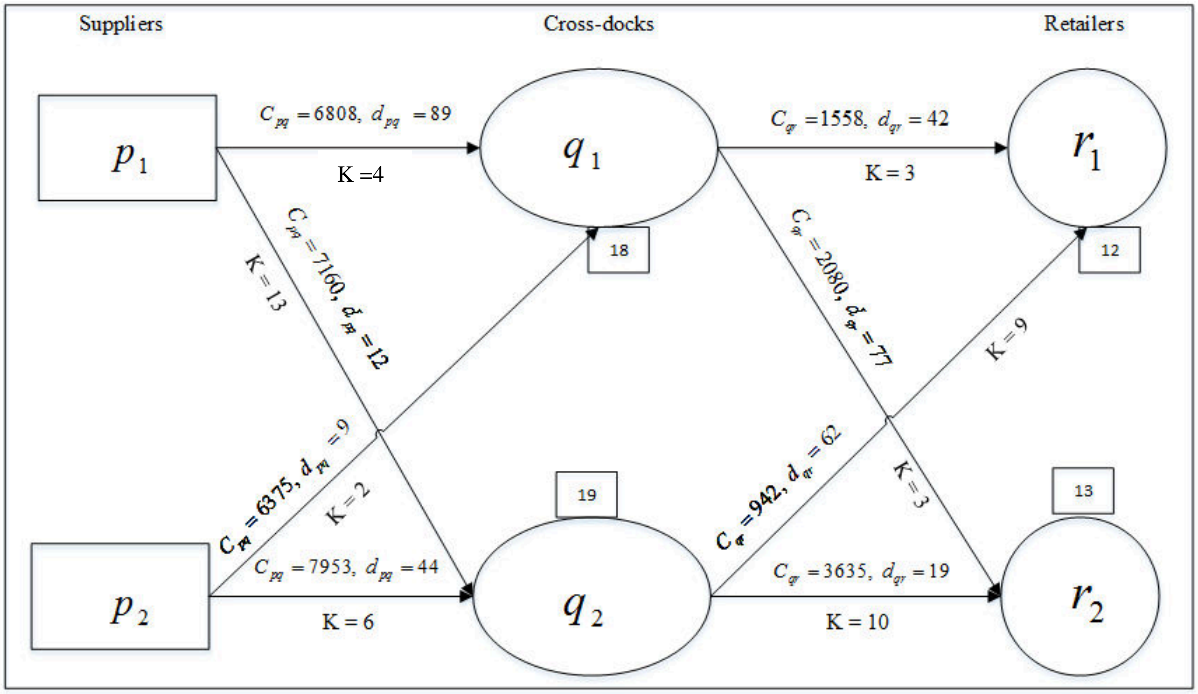
Suppliers
Cross-docks
Retailers
p_1
q_1
r_1
C_{pq} =6808, d_{pq} =89
K =4
C_{qr} =1558, d_{qr} = 42
K = 3
C_{pq} = 7160, d_{pq} =12
K = 13
18
C_{qr} = 2080, d_{qr} =77
12
K = 9
C_{pq} =6375, d_{pq} =9
K = 2
19
C_{qr} =942, d_{qr} = 62
K = 3
13
C_{pq} =7953, d_{pq} =44
K = 6
C_{qr} =3635, d_{qr} =19
K = 10
p_2
q_2
r_2

Figure 5. Pictorial representation of the dynamics of case 1 (2-2-1-2)

Figure 5 Alt text: Supply chain network showing the values of decision variables.

Table 5. Number of open cross-docks at different demand conditions

	Demand conditions					
	[5,15]		*[20,100]*		*[100,1000]*	
Case Type	**Available Cross docks**	**Open Cross docks**	**Available Cross docks**	**Open Cross docks**	**Available Cross docks**	**Open Cross docks**
Case 1 (2-2-1-2)	2	1	2	1	2	2
Case 2 (2-3-1-3)	3	1	3	2	3	3
Case 4 (2-3-1-10)	10	3	10	6	10	9
Case 13 (2-3-1-50)	50	3	50	26	50	42

5.4. Contribution to research and practice

The influence of carbon tax (cost) on overall supply chain costs is studied while designing a sustainable freight transportation network with cross-docks. It was observed that, although the carbon emission cost is much less compared to overall costs, it plays a vital role by encouraging decision-makers to consider them for meeting sustainability targets of the individual firm and the associated network. With stricter government regulations and the adverse impact of climate change, companies can afford to pay this nominal increase in costs to limit their carbon footprint in the environment. The research provides three significant contributions to the theory. Firstly, a sustainable model for a capacitated cross-docking system with varying (consisting of multiple suppliers and retailers) supply and demand conditions was developed, directly contributing to the theory development in sustainable freight transportation. Secondly, the contextual gap in utilising cross-docking systems for improving sustainability in freight transportation networks ((Wu et al. 2015) is bridged following the development of capacitated model considering multiple case scenarios. Thirdly, the research provides a methodological contribution by applying and validating the appropriateness of the TLSAVNS algorithm for complex supply chain problems. To the best of the authors' knowledge, the TLSAVNS algorithm, first proposed by Li and Tian (2016), has not been applied to or tested in the supply chain context. The carbon tax (cost) is an essential consideration for sustainable freight transportation due to the increased reverse logistics activities (Sarkar et al., 2016; Ghadge et al., 2016). The findings obtained from the study further strengthen the need for applying a carbon tax policy for building a sustainable freight transportation network. Researchers and practitioners are focused on finding emerging solutions for minimising negative externalities in the design of transportation systems (Demir et al., 2019). The study proves that the cross-docking strategy can support carbon emissions containment in freight transportation networks under varying supply chain nodes and demand scenarios.

The research also contributes to logistics and supply chain practice. The developed freight transportation network model supports freight distribution managers, particularly in understanding the usefulness of cross-docking systems in reducing carbon emissions and other operational costs, as it helps to reduce inventory holding and transportation costs. For the case company in India, this study helps to plan their cross-docking numbers and locations for a varying set of suppliers and retailers for different supply and demand settings. This model could be applied to other developing countries cases also considering their logistics and transportation infrastructure, technologies, government regulations and other policies. Developed case scenarios based on the combination of suppliers, retailers, cross-docks, and products, help in making evidence-based, robust decisions related to several open cross-docks for varying capacity. The number of open cross-docks increases with higher demand distribution, which implies it is better to opt for direct shipment for small demand size; but as the demand size increases, it is more economical to use a cross-dock logistics network. Also, it is found that the capacity of the cross-docks is closely related to the number of open cross-docks, as with the increases in the capacity, the number of open cross-docks decreases. These results can be used for a trade-off between the two, in terms of cost incurred to determine whether it is profitable to establish a new cross-dock or to invest in a cross-dock of higher capacity. The relational dynamics between supply and demand for open cross-docks is another useful insight for freight transportation practitioners. The results are likely to bring meaningful savings in the operational and regulatory costs associated with cross-docking systems and broader sustainable freight transportation.

6. Conclusion and future scope of research

This research study is inspired from the realistic problem of a major retail distribution company based in India. The proposed model for a cross-docking system computed the total cost of the supply chain network considering several suppliers and retailers. A mixed-integer non-linear programming model is formulated to address the sustainable three-echelon supply chain network problem for deterministic demand and supply scenarios. Due to the inherent complexity, the mathematical model is solved by applying two neighbourhood search algorithms. Unlike the VNS algorithm, which uses a pre-determined sequence of neighbourhoods, the TLSAVNS algorithm applies a self-adaptive neighbourhood search strategy based on the selection probability of each neighbourhood. Multiple findings are reported after solving several cases using both algorithms. It is also found that the self-adaptive approach of the TLSAVNS algorithm is robust in providing near-optimal solutions in less computational time. The searching procedure of the TLSAVNS algorithm helps to smoothly escape from the local optima and promptly achieve a near-optimal solution, which can be consistently witnessed from the results.

Like any other study, this study has some limitations which can be addressed in the future. The heterogeneous capacitated vehicles and their limited availability can be included in new models. Although an attempt is made to accommodate important costs, several additional costs (such as administrative and technology costs) and the time dimension are challenging to incorporate in the mathematical formulation (Agamez-Arias and Moyano-Fuentes, 2017); thus, were not considered while modelling this problem. Future research can investigate accommodating such missing variables/factors. Another exciting direction for this research could be the consideration of the stochastic demand, supply, and travel time to make the model robust in tackling uncertain scenarios. The integration of vehicle routing and scheduling could be another interesting future search avenue that helps to optimise the overall network cost. The environmental impact of facility location and inventory could be added to the developed model. The multi-objective model can be formulated with the consideration of the third dimension of sustainability, i.e., social factors in the form of employment opportunities, the impact of traffic congestions, noise pollution and accidents. Finally, scholars can verify the applicability of the proposed TLSAVNS algorithm by solving different type of supply chain network problems.

DAS statement

'The authors confirm that the data supporting the findings of this study are available within the article'.

References

Abad, H. Kargari Esfand, Behnam Vahdani, M. Sharifi, and F. Etebari. 2018. "A bi-objective model for pickup and delivery pollution-routing problem with integration and consolidation shipments in cross-docking system." *Journal of Cleaner Production* 193: 784-801.

Abouee-Mehrizi, Hossein, Oded Berman, and M. Reza Baharnemati. 2014. "Designing production-inventory-transportation systems with capacitated cross-docks." *Transportation Science* 48 (1): 121-135.

Agamez-Arias, Anny del Mar, and José Moyano-Fuentes. 2017. "Intermodal Transport in Freight Distribution: A Literature Review." *Transport Reviews* 37 (6): 782–807.

Alguwaizani, Abdulrahman, Pierre Hansen, Nenad Mladenović, and Eric Ngai. 2011. "Variable Neighborhood Search for Harmonic Means Clustering." *Applied Mathematical Modelling* 35 (6): 2688–2694.

Benrqya, Yassine. 2019. "Costs and Benefits of Using Cross-Docking in the Retail Supply Chain: A Case Study of an FMCG Company." *International Journal of Retail and Distribution Management* 47 (4): 412–432.

Berman, Oded, and Qian Wang. 2006. "Inbound Logistic Planning: Minimising Transportation and Inventory Cost." *Transportation Science* 40 (3): 287–299.

Bertazzi, Luca, Adamo Bosco, and Demetrio Laganà. 2016. "Min-Max Exact and Heuristic Policies for a Two-Echelon Supply Chain with Inventory and Transportation Procurement

Decisions.” *Transportation Research Part E: Logistics and Transportation Review* 93: 57–70.

Chargui, Tarik, Abdelghani Bekrar, Mohamed Reghioui, and Damien Trentesaux. 2019. "Multi-Objective sustainable truck scheduling in a rail–road physical internet cross-docking hub considering energy consumption." *Sustainability* 11 (11): 3127.

Chen, Mu Chen, Yu Hsiang Hsiao, Reddivari Himadeep Reddy, and Manoj Kumar Tiwari. 2016. “The Self-Learning Particle Swarm Optimization Approach for Routing Pickup and Delivery of Multiple Products with Material Handling in Multiple Cross-Docks.” *Transportation Research Part E: Logistics and Transportation Review* 91: 208–226.

De Campos, Renan Stenico, Alexandre Tadeu Simon, and Felipe De Campos Martins. 2019. “Assessing the Impacts of Road Freight Transport on Sustainability: A Case Study in the Sugar-Energy Sector.” *Journal of Cleaner Production* 220: 995–1004.

De, A., Wang, J. and Tiwari, M.K., 2019a. Hybridizing basic variable neighborhood search with particle swarm optimization for solving sustainable ship routing and bunker management problem. *IEEE Transactions on Intelligent Transportation Systems*, *21*(3): 986–997.

De, Arijit, Alok Choudhary, and Manoj Kumar Tiwari. 2019b. "Multi-objective Approach for Sustainable Ship Routing and Scheduling with Draft Restrictions." *IEEE Transactions on Engineering Management* 66 (1): 35–51.

De, Arijit, D. G. Mogale, Mengdi Zhang, Saurabh Pratap, Sri Krishna Kumar, and George Q. Huang. 2020. “Multi-Period Multi-Echelon Inventory Transportation Problem Considering Stakeholders Behavioural Tendencies.” *International Journal of Production Economics* 225.

Demir, Emrah, Martin Hrušovský, Werner Jammernegg, and Tom Van Woensel. 2019. “Green Intermodal Freight Transportation: Bi-Objective Modelling and Analysis.” *International Journal of Production Research* 57 (19): 6162–6180.

Dente, Sébastien M.R., and Lóránt A. Tavasszy. 2018. “Impacts of Trade Related Sustainability Strategies on Freight Transportation: Modelling Framework and Application for France.” *Transportation Research Part D: Transport and Environment* 58: 308–319.

Diabat, Ali, and Mohammed Al-Salem. 2015. “An Integrated Supply Chain Problem with Environmental Considerations.” *International Journal of Production Economics* 164: 330–338.

Ding, Huiping, Qilan Zhao, Zhirong An, and Ou Tang. 2016. “Collaborative Mechanism of a Sustainable Supply Chain with Environmental Constraints and Carbon Caps.” *International Journal of Production Economics* 181: 191–207.

Dulebenets, Maxim A. 2018. "A diploid evolutionary algorithm for sustainable truck scheduling at a cross-docking facility." *Sustainability* 10 (5): 1333.

Dulebenets, Maxim A. 2019. “A Delayed Start Parallel Evolutionary Algorithm for Just-in-Time Truck Scheduling at a Cross-Docking Facility.” *International Journal of Production Economics* 212: 236–258.

Fahimnia, Behnam, Joseph Sarkis, Alok Choudhary, and Ali Eshragh. 2015. “Tactical Supply Chain Planning under a Carbon Tax Policy Scheme: A Case Study.” *International Journal of Production Economics* 164: 206–215.

Garg, Amit, P. R. Shukla, Bhushan Kankal, and Diptiranjan Mahapatra. 2017. "CO2 emission in India: trends and management at sectoral, sub-regional and plant levels." *Carbon Management* 8 (2): 111-123.

Gaudioso, Manlio, Maria Flavia Monaco, and Marcello Sammarra. 2021. "A Lagrangian heuristics for the truck scheduling problem in multi-door, multi-product Cross-Docking with constant processing time." Omega 101: 102255.

Gawankar, Shradha A., Angappa Gunasekaran, and Sachin Kamble. 2020. “A Study on Investments in the Big Data-Driven Supply Chain, Performance Measures and Organisational Performance in Indian Retail 4.0 Context.” *International Journal of Production Research* 58 (5): 1574–1593.

Ghadge, Abhijeet, Qifan Yang, Nigel Caldwell, Christian König, and Manoj Kumar Tiwari. 2016. “Facility Location for a Closed-Loop Distribution Network: A Hybrid Approach.” *International Journal of Retail and Distribution Management* 44 (9): 884–902.

Ghadge, Abhijeet, Wurtmann, Hendrik, and Seuring Stefan. 2020. Managing climate change risks in global supply chains: a review and research agenda. *International Journal of Production Research*, 58(1), 44-64.

Goodarzi, Asefeh Hasani, and Seyed Hessameddin Zegordi. 2016. “A Location-Routing Problem for Cross-Docking Networks: A Biogeography-Based Optimisation Algorithm.” *Computers and Industrial Engineering* 102: 132–146.

Goodarzi, Asefeh Hasani, Seyed Hessameddin Zegordi, Gülgün Alpan, Isa Nakhai Kamalabadi, and Ali Husseinzadeh Kashan. 2020. *Reliable Cross-Docking Location Problem under the Risk of Disruptions*. *Operational Research*. Springer Berlin Heidelberg. doi:10.1007/s12351-020-00583-5.

Goswami, Mohit, Arijit De, Muhammad Khoirul Khakim Habibi, and Yash Daultani. 2020. “Examining Freight Performance of Third-Party Logistics Providers within the Automotive Industry in India: An Environmental Sustainability Perspective.” *International Journal of Production Research* 7543.

Government of India. 2015. "India’s intended nationally determined contribution: Working towards climate justice." Vikaspedia.

Govindan, Kannan, Ahmad Jafarian, and Vahid Nourbakhsh. 2019. “Designing a Sustainable Supply Chain Network Integrated with Vehicle Routing: A Comparison of Hybrid Swarm Intelligence Metaheuristics.” *Computers and Operations Research* 110: 220–235.

Gruler, Aljoscha, Javier Panadero, Jesica de Armas, José A.Moreno Pérez, and Angel A. Juan. 2020. “A Variable Neighborhood Search Simheuristic for the Multiperiod Inventory Routing Problem with Stochastic Demands.” *International Transactions in Operational Research* 27 (1): 314–335.

Hansen, Pierre, Nenad Mladenović, and José A. Moreno Pérez. 2010. "Variable neighbourhood search: methods and applications." *Annals of Operations Research* 175 (1): 367-407.

Hansen, Pierre, and Nenad Mladenović. 2014. "Variable neighborhood search." In Search methodologies, 313-337. Springer, Boston.

Jarboui, B., Derbel, H., & Mladenović, N. (2013). Variable neighborhood search for location routing, *Computers & Operations Research, 40*(1): 47–57.

Jin, Mingzhou, Nelson A. Granda-Marulanda, and Ian Down. 2014. “The Impact of Carbon Policies on Supply Chain Design and Logistics of a Major Retailer.” *Journal of Cleaner Production* 85: 453–461.

Khanna Sundeep. 2020. https://www.livemint.com/opinion/online-views/india-s-economy-needs-the-retail-sector-to-grow-11588785884086.html.

Kiani Mavi, Reza, Mark Goh, Neda Kiani Mavi, Ferry Jie, Kerry Brown, Sharon Biermann, and Ahmad A Khanfar. 2020. "Cross-docking: A systematic literature review." *Sustainability* 12 (11): 4789.

Konur, Dinçer, and Brian Schaefer. 2014. “Integrated Inventory Control and Transportation Decisions under Carbon Emissions Regulations: LTL vs. TL Carriers.” *Transportation Research Part E: Logistics and Transportation Review* 68 (2014): 14–38.

Küçükoğlu, İlker, and Nursel Öztürk. 2017. “Two-Stage Optimisation Method for Material Flow and Allocation Management in Cross-Docking Networks.” *International Journal of Production Research* 55 (2): 410–429.

Kumar, Aalok, and Ramesh Anbanandam. 2019. "Multimodal freight transportation strategic network design for sustainable supply chain: an or prospective literature review." *International Journal of System Dynamics Applications* 8 (2): 19-35.

Kumar, Aalok, and Ramesh Anbanandam. 2020. “Assessment of Environmental and Social Sustainability Performance of the Freight Transportation Industry: An Index-Based Approach.” *Transport Policy*, no. xxxx: doi:10.1016/j.tranpol.2020.01.006.

Kumar, Akhilesh, Martina Calzavara, Nagendra R. Velaga, Alok Choudhary, and Ravi Shankar. 2019. “Modelling and Analysis of Sustainable Freight Transportation.” *International Journal of Production Research* 57 (19): 6086–6089. doi:10.1080/00207543.2019.1642689.

Kuo, Yiyo, and Chi Chang Wang. 2012. “A Variable Neighborhood Search for the Multi-Depot Vehicle Routing Problem with Loading Cost.” *Expert Systems with Applications* 39 (8): 6949–6954.

Ladier, Anne-Laure, and Gülgün Alpan. 2016. "Cross-docking operations: Current research versus industry practice." *Omega* 62: 145-162.

Lee, Y. C.E., Chi Kin Chan, A. Langevin, and H. W.J. Lee. 2016. “Integrated Inventory-Transportation Model by Synchronising Delivery and Production Cycles.” *Transportation Research Part E: Logistics and Transportation Review* 91: 68–89.

Li, Kun, and Huixin Tian. 2016. “A Two-Level Self-Adaptive Variable Neighborhood Search Algorithm for the Prize-Collecting Vehicle Routing Problem.” *Applied Soft Computing Journal* 43: 469–479.

Liotta, Giacomo, Giuseppe Stecca, and Toshiya Kaihara. 2015. "Optimisation of freight flows and sourcing in sustainable production and transportation networks." *International Journal of Production Economics* 164: 351-365.

M’Hallah, R., Alkandari, A., & Mladenovic, N. (2013). Packing unit spheres into the smallest sphere using VNS and NLP. *Computers and Operations Research*, *40*(2): 603–615.

Mahapatra, Diptiranjan, Rajesh Katiyar, Ratri Parida, and Divesh Kumar. 2020. “A Fuzzy Multi-Criteria Approach for Evaluating the Contribution of Freight Transportation towards India’s Nationally Determined Contributions (NDCs).” *International Journal of Production Research* 7543.

Maiyar, Lohithaksha M., and Jitesh J. Thakkar. 2019. "Modelling and analysis of intermodal food grain transportation under hub disruption towards sustainability." International Journal of Production Economics 217: 281-297.

Maknoon, Yousef, and Gilbert Laporte. 2017. “Vehicle Routing with Cross-Dock Selection.” *Computers and Operations Research* 77: 254–266.

Menendez, B., Pardo, E.G., Alonso-Ayuso, A., Molina, E., and Duarte, A. 2017. “Variable Neighborhood Search strategies for the Order Batching Problem.” *Computers & Operations Research* 78: 500-512

Mladenović, N., and Hansen, P. 1997. Variable neighborhood search. Computers Operations Research. 24: 1097–1100.

Mladenović, N., Todosijević, R., & Urošević, D. (2014). Two level General variable neighborhood search for Attractive traveling salesman problem. *Computers and Operations Research*, *52*: 341–348.

Mogale, D. G., Sri Krishna Kumar, and Manoj Kumar Tiwari. 2018. “An MINLP Model to Support the Movement and Storage Decisions of the Indian Food Grain Supply Chain.” *Control Engineering Practice* 70: 98–113.

Mogale, D. G., Sri Krishna Kumar, and Manoj Kumar Tiwari. 2020. “Green Food Supply Chain Design Considering Risk and Post-Harvest Losses: A Case Study.” *Annals of Operations Research*, 295: 257-284.

Moghadam, S. Shahin, SMT Fatemi Ghomi, and Behrooz Karimi. 2014. "Vehicle routing scheduling problem with cross docking and split deliveries." *Computers & chemical engineering* 69: 98-107.

Pan, Fei, Wei Zhou, Tijun Fan, Shuxia Li, and Chong Zhang. 2021. "Deterioration rate variation risk for sustainable cross-docking service operations." *International Journal of Production Economics* 232: 107932.

Pathak, Devendra Kumar, Lakshman S. Thakur, and Shams Rahman. 2019. “Performance Evaluation Framework for Sustainable Freight Transportation Systems.” *International Journal of Production Research* 57 (19): 6202–6222.

Rezaei, Saeid, and Amirsaman Kheirkhah. 2017. "Applying forward and reverse cross-docking in a multi-product integrated supply chain network." *Production Engineering* 11 (4): 495-509.

Rezaei, Saeid, and Amirsaman Kheirkhah. 2018. "A comprehensive approach in designing a sustainable closed-loop supply chain network using cross-docking operations." *Computational and Mathematical Organization Theory* 24 (1): 51-98.

Rostami, Ahmadreza, Mohammad Mahdi Paydar, and Ebrahim Asadi-Gangraj. 2020. “A Hybrid Genetic Algorithm for Integrating Virtual Cellular Manufacturing with Supply Chain Management Considering New Product Development.” *Computers and Industrial Engineering* 145: 106565. doi:10.1016/j.cie.2020.106565.

Samà, Marcella, Francesco Corman, and Dario Pacciarelli. 2017. "A variable neighbourhood search for fast train scheduling and routing during disturbed railway traffic situations." *Computers & Operations Research* 78: 480-499.

Salhi, S., Imran, A., & Wassan, N. A. (2014). The multi-depot vehicle routing problem with heterogeneous vehicle fleet: Formulation and a variable neighborhood search implementation. *Computers and Operations Research*, *52*: 315–325.

Sarkar, Biswajit, Baishakhi Ganguly, Mitali Sarkar, and Sarla Pareek. 2016. "Effect of variable transportation and carbon emission in a three-echelon supply chain model." *Transportation Research Part E: Logistics and Transportation Review* 91: 112-128.

Schaefer, Brian, and Dinçer Konur. 2015. "Economic and environmental considerations in a continuous review inventory control system with integrated transportation decisions." *Transportation Research Part E: Logistics and Transportation Review* 80: 142-165.

Şevkli, Aişe Zülal, and Bekir Güler. 2017. "A Multi-Phase Oscillated Variable Neighbourhood Search Algorithm for a Real-World Open Vehicle Routing Problem." *Applied Soft Computing Journal* 58: 128–144.

Shahmardan, Amin, and Mohsen S. Sajadieh. 2020. "Truck Scheduling in a Multi-Door Cross-Docking Center with Partial Unloading – Reinforcement Learning-Based Simulated Annealing Approaches." *Computers and Industrial Engineering* 139: 106134.

Shankar, Ravi, Devendra Kumar Pathak, and Devendra Choudhary. 2019. "Decarbonising Freight Transportation: An Integrated EFA-TISM Approach to Model Enablers of Dedicated Freight Corridors." *Technological Forecasting and Social Change* 143: 85–100.

Steadieseifi, M., N. P. Dellaert, W. Nuijten, T. Van Woensel, and R. Raoufi. 2014. "Multimodal Freight Transportation Planning: A Literature Review." *European Journal of Operational Research* 233 (1): 1–15.

Stenger, Andreas, Daniele Vigo, Steffen Enz, and Michael Schwind. 2013. "An adaptive variable neighborhood search algorithm for a vehicle routing problem arising in small package shipping." *Transportation Science* 47 (1): 64-80.

Sze, Jeeu Fong, Said Salhi, and Niaz Wassan. 2017. "The Cumulative Capacitated Vehicle Routing Problem with Min-Sum and Min-Max Objectives: An Effective Hybridisation of Adaptive Variable Neighbourhood Search and Large Neighbourhood Search." *Transportation Research Part B: Methodological* 101:162–184.

Tang, Lixin, and Xianpeng Wang. 2006. "Iterated local search algorithm based on very large-scale neighborhood for prize-collecting vehicle routing problem." *The International Journal of Advanced Manufacturing Technology* 29 (11-12): 1246-1258.

Tirkolaee, Erfan Babaee, Alireza Goli, Amin Faridnia, Mehdi Soltani, and Gerhard-Wilhelm Weber. 2020. "Multi-objective optimization for the reliable pollution-routing problem with cross-dock selection using Pareto-based algorithms." *Journal of Cleaner Production* 276: 122927.

Todosijević, Raca, Saïd Hanafi, Dragan Urošević, Bassem Jarboui, and Bernard Gendron. 2017. "A general variable neighborhood search for the swap-body vehicle routing problem." *Computers & Operations Research* 78: 468-479.

Urzúa-Morales, Juan Guillermo, Juan Pedro Sepulveda-Rojas, Miguel Alfaro, Guillermo Fuertes, Rodrigo Ternero, and Manuel Vargas. 2020. "Logistic modeling of the last mile: Case study Santiago, Chile." *Sustainability* 12 (2): 648.

Vahdani, Behnam. 2019. "Assignment and scheduling trucks in cross-docking system with energy consumption consideration and trucks queuing." *Journal of Cleaner Production* 213: 21-41.

Vahdani, Behnam, and Mostafa Zandieh. 2010. "Scheduling trucks in cross-docking systems: Robust meta-heuristics." *Computers & Industrial Engineering* 58 (1): 12-24.

Van Belle, Jan, Paul Valckenaers, and Dirk Cattrysse. 2012. "Cross-docking: State of the art." *Omega* 40 (6): 827-846.

Vidal, Thibaut, Nelson Maculan, Luiz Satoru Ochi, and Puca Huachi Vaz Penna. 2016. "Large neighborhoods with implicit customer selection for vehicle routing problems with profits." *Transportation Science* 50 (2): 720-734.

Wassan, Naveed, Niaz Wassan, Gábor Nagy, and Saïd Salhi. 2017. "The multiple trip vehicle routing problem with backhauls: Formulation and a two-level variable neighbourhood search." *Computers & Operations Research* 78: 454-467.

World Health Organisation (WHO). 2020. https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries.

Wu, Ting, Houcai Shen, and Cheng Zhu. 2015. "A Multi-Period Location Model with Transportation Economies-of-Scale and Perishable Inventory." *International Journal of Production Economics* 169: 343–349.

Xiao, Yiyong, Renqian Zhang, Qiuhong Zhao, Ikou Kaku, and Yuchun Xu. 2014. "A Variable Neighborhood Search with an Effective Local Search for Uncapacitated Multilevel Lot-Sizing Problems." *European Journal of Operational Research* 235 (1): 102–114.

Yang, P., Miao, L., Xue, Z., & Ye, B. (2015). Variable neighborhood search heuristic for storage location assignment and storage/retrieval scheduling under shared storage in multi-shuttle automated storage/retrieval systems. *Transportation Research Part E: Logistics and Transportation Review*, *79*: 164-177.

Yazdani, Morteza, Dragan Pamucar, Prasenjit Chatterjee, and Shankar Chakraborty. 2020. "Development of a Decision Support Framework for Sustainable Freight Transport System Evaluation Using Rough Numbers." *International Journal of Production Research* 58 (14): 4325–4351.

Yu, Shiwei, Shuhong Zheng, Shiwei Gao, and Juan Yang. 2017. "A Multi-Objective Decision Model for Investment in Energy Savings and Emission Reductions in Coal Mining." *European Journal of Operational Research* 260 (1): 335–347.

Yu, Vincent F., Parida Jewpanya, and Voratas Kachitvichyanukul. 2016. "Particle swarm optimization for the multi-period cross-docking distribution problem with time windows." *International Journal of Production Research* 54 (2): 509-525.

Zhao, Shi Tao, Kan Wu, and Xue Ming Yuan. 2016. “Optimal Integer-Ratio Inventory Coordination Policy for an Integrated Multi-Stage Supply Chain.” *Applied Mathematical Modelling* 40 (5–6). Elsevier Inc.: 3876–3894.