Ade Irawan, Chandra, Dan-Asabe Abdulrahman, Muhammad, Salhi, Said and Luis, Martino (2022) ***An efficient matheuristic algorithm for bi-objective sustainable closed-loop supply chain networks.*** **IMA Journal of Management Mathematics, 33 (4). pp. 603-636. ISSN 1471-678X.**

IMA Journal of Management Mathematics (2022) **33**, 603–636
https://doi.org/10.1093/imaman/dpac003
Advance Access publication on 19 May 2022

An efficient matheuristic algorithm for bi-objective sustainable closed-loop supply chain networks

Chandra Ade Irawan
Nottingham University Business School China, University of Nottingham Ningbo China, 199 Taikang East Road, Ningbo 315100, China

Muhammad Dan-Asabe Abdulrahman
School of Accounting, Information Systems and Supply Chain, College of Business, RMIT University, Melbourne, VIC 3000, Australia

Said Salhi
Centre for Logistics and Heuristic Optimization (CLHO), Kent Business School, University of Kent, Kent CT2 7PE, UK

and

Martino Luis[†]
College of Engineering, Mathematics, and Physical Sciences, University of Exeter, Exeter EX4 4QF, UK
[†]Corresponding author. Email: m.luis@exeter.ac.uk

[Received on 13 March 2021; accepted on 12 April 2022]

This paper develops an optimization model for a sustainable closed-loop supply chain network with two conflicting objectives, namely, the minimization of the total logistic costs and the total amount of carbon emissions. The first objective relates to financial benefits, whereas the second represents the wider goal of guaranteeing cleaner air and hence a greener and healthier planet. The problem is first modelled as a mixed integer linear programming based-model. The aim is to determine the location of distribution centres and recycling centres, their respective numbers and the type of vehicles assigned to each facility. Vehicle type consideration, not commonly used in the literature, adds another dimension to this practical and challenging logistic problem. A matheuristic using compromise programming is put forward to tackle the problem. The proposed matheuristic is evaluated using a variety of newly generated datasets which produces compromise solutions that demonstrate the importance of an appropriate balance of both objective functions. The robustness analysis considering fluctuations in customer demand is assessed using Monte Carlo simulation. The results show that if the standard deviation of the demand falls within 10% of its average, the unsatisfied demand is insignificant, thus demonstrating the stability of supply chain configuration. This invaluable information is key towards helping senior management make relevant operational and strategic decisions that could impact on both the sustainability and the resilience of their supply chain networks.

Keywords: closed-loop supply chain network; bi-objective; sustainability; matheuristic.

1. Introduction

The needs for a well-designed closed-loop supply chain (CLSC) have gained prominence in recent years. Several reasons are responsible for this renewed push for CLSC ranging from economic considerations, government legislations, increased customers' expectations for better business stewardship and global push for circular economy (Kazancoglu *et al.*, 2018; Kusakci *et al.*, 2019). The singular most important purpose of these network designs is for efficient and cost-effective approaches to achieve sustainability in the company's environmental and economic performances (Asgari *et al.*, 2016; Dai & Zheng, 2015).

In CLSC, the flow of products is between two markets at both ends of the supply chain, with the forward flow represents the market at the beginning of the CLSC while the reverse flow represents the market at the end of the CLSC. The reverse flow transports used, damage, unsold and/or end-of-life products back to the manufacturers. This flow facilitates reprocessing of such returns with value-added methods based on material recovery and reuse, recycling, repair, refurbishing and remanufacturing. A well-structured CLSC system therefore facilitates the attainment of a circular economy that focuses on recycling, conservation of scarce resources and waste transformation into wealth generation (Diabat & Al-Salem, 2015; Ghisellini *et al.*, 2016). It also addresses vital operational and strategic decisions on inventory positioning/ management and optimal vehicle routing to achieve reduced costs and increased customer service levels (Asl-Najafi *et al.*, 2015). Furthermore, a well-designed CLSC system enables low cost determination of optimal product flow quantity within a network (Dai & Zheng, 2015; Kadambala *et al.*, 2017), with the reverse strategy providing a significant cost savings and reduced operational greenhouse gas emissions (GHGs) considerably (Abdulrahman *et al.*, 2014).

Despite the established benefits, the design and the implementation of a cost-effective CLSC system are not easy to achieve. The difficulties lie in the need to simultaneously consider both forward and reverse supply networks to obtain desired results (Pishvaee & Razmi, 2012). In other words, solving forward supply chain and reverse logistics separately as traditionally done will only results in sub-optimal achievements of reduced cost, transport pollution reduction and optimal flow quantities. Furthermore, senior management often face a challenging decision when attempting to resolve the multiple conflicting decision choices within a CLSC design solution. Such dilemma include a company's desired level of profitability, demand variability, rate of product return and the firm's desired position in terms of social responsibility, among others (Elhedhli & Merrick, 2012; Kadambala *et al.*, 2017; Pan *et al.*, 2013).

Several extant studies have attempted to address these challenging issues by investigating CLSC designs from different perspectives. The majority of them focused on CLSC designs from a sustainability perspective (Atabaki *et al.*, 2020; Devika *et al.*, 2014; Fahimnia *et al.*, 2013; Mota *et al.*, 2015; Talaei *et al.*, 2016; Tiwari *et al.*, 2016; Tosarkani & Amin, 2018). A major problem with these and similar studies is the lack of a holistic approach, which focuses on the challenging issue of solving the forward and the reverse supply chain (SC) networks simultaneously. A similar issue of ignoring routing when locating facilities was initially demonstrated by Salhi & Rand (1989). This issue of sub-optimality has attracted a lot of attention and research in the area of location-routing problem. We believe that this paper follows a similar line of research but for the case of CLSC instead. Furthermore, all extant studies assumed only a single type of vehicle for the bidirectional flow of products in their designs. This is a fundamental omission given the acknowledged environmental impact of transportation through GHG (Dekker *et al.*, 2012; Elhedhli & Merrick, 2012; Pan *et al.*, 2013). All these are in addition to the limited number of facilities considered in most of the designs of the extant studies.

To address these gaps and facilitate informed decision-making, we develop a generic optimisation model for the sustainable CLSC network design (SCLCND). Our aim is to determine the number and

locations of distribution centres (DCs) and recycling centres (RCs) together with their capacity, transportation modes for each facility and the bidirectional flows of products in order to minimize the total costs and carbon emissions (CO_2) generated. We adopt a compromise programming (CP) methodology to obtain compromise solutions that deal with the bi-objective problem, namely, minimizing the amount of CO_2 emissions and the total operational cost.
The contributions of this study are four-fold.

- The development of an efficient integrated bi-objective model based on CP for designing a sustainable CLSC network.
- The construction and analysis of an effective matheuristic which integrates an aggregation technique, an exact method and a local search.
- The ability to solve large newly constructed datasets, which can be used for future benchmarking purposes.
- The flexibility in investigating both scenario analysis and Monte Carlo simulation that would assist senior management when making their strategic decisions that could impact on both the sustainability and the robustness of their supply network.

The rest of the study is organized as follows. Section 2 summarized the literature review. Section 3 provides a description of the proposed mathematical model, while Section 4 covers the solution method. The matheuristic is presented in Section 5. The computational results, including the newly constructed datasets, a scenario analysis and a simulation experiment are provided in Section 6. A summary of the findings and some future research avenues are highlighted in the final section.

2. Literature review

In the last two decades, studies on sustainable CLSC developments have received extensive interest due to growing resource scarcity, environmental concerns and huge financial impact of product returns.To mitigate these negative impacts, extant studies in the CLSC network design have examined, amongst other aspects, facility location and allocation, capacities of facilities, production planning, inventory, transportation and environmental impact (Altmann & Bogaschewsky, 2014; Ghahremani-Nahr *et al.*, 2019; Kang *et al.*, 2017; Kaya & Urek, 2016; Subramanian *et al.*, 2013). For a comprehensive review on the CLSC, the reader can refer to the works of Govindan *et al.* (2015, 2017), and recently, Oliveira & Machado (2021) presented a systematic review on optimisation techniques applied in the CLSC.

Fleischmann *et al.* (2001) was among the first to study a generic uncapacitated CLSC model with product recovery facilities as a mixed integer linear programming (MILP). The model was assessed using two case studies concerning copier remanufacturing and paper recycling, respectively and was solved using commercial solver package. Lu & Bostel (2007) developed a 0-1 mixed integer programming (MIP) for a remanufacturing closed-loop SC network and a lagrangian-based heuristic was designed to tackle the problem. Demirel & Gökçen (2008) formulated a remanufacturing system in a CLSC as MILP and solved the problem using genetic algorithm (GA) to get quantities of manufactured and remanufactured products while finding the locations of disassembly, collection and distribution centres. Yi *et al.* (2016) extended the study of a CLSC with remanufacturing facilities and a hybrid GA was applied to solve the problem in finding the location of various centre, flows of used products, components and remanufactured products.

Salema *et al.* (2009) put forward a MILP for a multi-product and multi-period CLSC model by embedding strategic and tactical location-allocation decisions where the model was solved using branch and bound technique. Wang & Hsu (2010) constructed an integer linear programming (ILP) for a CLSC logistics system and developed a spanning tree based GA to deal with the model.

CLSC network design problems have also been extended to include various aspects in practice, such as third party logistics (3PL) companies, recyclable products, life cycle assessment principles, pricing decisions, and uncertainty and product return, among others. Ko & Evans (2007) integrated 3PL operators into a CLSC model and applied genetic algorithm to tackle the model. In a similar fashion, Li *et al.* (2018) studied a CLSC model with location-inventory problem where the logistic service was outsourced to 3PL. A hybrid heuristic based on the improved hybrid differential evolution algorithm and GA was put forward to solve the proposed model.

Chaabane *et al.* (2012) investigated a sustainable CLSC that considers Life Cycle Assessment (LCA) principles and utilised an optimisation software package to solve the problem. Pishvaee *et al.* (2010) introduced a multi-objective fuzzy MIP for designing an environmental CLSC network. LCA method was applied to assess and quantify the environmental influence of the network.

Kalaitzidou *et al.* (2015) put forward a MILP model for a CLSC network design with recyclable products and multifunctional nodes. A branch and bound methodology was implemented to solve the proposed MILP model using a real case study from a Europe based consumer goods company. Kaya & Urek (2016) proposed a MINLP to incorporate pricing decisions into a CLSC network and introduced incentive values for the collection of right amount of recyclable products into the model. Three hybrid metaheuristics based on simulated annealing (SA), tabu search (TS), GA, and Variable Neighborhood Search (VNS) were designed as solution methods. Patne *et al.* (2018),extended the work of Kaya & Urek (2016) and developed an improved particle swarm optimisation algorithm to deal with the model. Atabaki *et al.* (2019) examined a CLSC with price-sensitive demand and put forward a priority-based firefly metaheuristic algorithm.

Pishvaee & Torabi (2010) discussed a bi-objective MIP formulation for a CLSC network with product return under uncertainty. To solve the proposed model, the authors put forward a multi-objective memetic algorithm with dynamic local search mechanism. Subramanian *et al.* (2013) presented a CLSC model by considering fixed charge for locating facilities and warehouse to organise uncertainty of product returns efficiently. A priority based simulated annealing was proposed as a solution procedure. Yadegari *et al.* (2019) hybridised a memetic algorithm with a multi-start SA algorithm to deal with location and product flow decisions in a multi-period CLSC network design. Zhen *et al.* (2019) built a two-stage stochastic MINLP for an integrated CLSC under uncertain demand and return. An improved TS algorithm was suggested to tackle the problem.

Many previous works have focused on an implementation of CLSC in various industries, such as computer products (Chen *et al.*, 2015; Kusumastuti *et al.*, 2008; Lee & Dong, 2008), glass (Devika *et al.*, 2014), tyre (Fathollahi-Fard *et al.*, 2018), battery (Tosarkani & Amin, 2018), bottled water (Papen & Amin, 2019), dairy (Gholizadeh *et al.*, 2021), and walnut (Salehi-Amiri *et al.*, 2021). Another variant of CLSC, known as green CLSC network models, has included the environmental element into the objective function; for example, in the works of Fahimnia *et al.* (2013), Altmann & Bogaschewsky (2014), Choudhary *et al.* (2015), Talaei *et al.* (2016), Tiwari *et al.* (2016), and Atabaki *et al.* (2020). A number of previous studies on the CLSC network design has integrated fuzzy environment into the model, such as Ramezani *et al.* (2014), Kang *et al.* (2017), Govindan & Soleimani (2017), Ghahremani-Nahr *et al.* (2019), and Nayeri *et al.* (2020).

Table 1 presents a summary of relevant previous works and highlights the distinctive aspects of the current study (last row in bold). The fourth column of Table 1 indicates the sustainability aspects used in the corresponding articles where Eco, Env, and Soc refer to the Economic, Environmental and Social aspects of sustainability respectively. The fifth column provides outputs or decision variables considered in each paper where L refers to facility location, AL indicates allocation, FC denotes facility capacity, TM represents transportation mode, TA stands for transportation amount, and ND expresses quantity of non-satisfied demand.

In this paper, an integrated SCLCND problem is investigated where a mixed integer linear programming (MILP) is constructed to determine the number and locations of DCs and RCs together with their capacity, transportation modes for each facility and the flows of products in order to deal with the total costs and carbon emissions (CO_2) produced. In addition, it is common in the literature, as earlier stated, that a facility uses only one type of vehicle to transport their products. Here, we extend the problem so that the optimal vehicle type can also be determined for each facility. Although this increases the complexity of the problem, in practice, it is important to consider this activity as part of the overall company strategy. Moreover, a combination of single- and multi-sources allocations is applied in the model. The multi-source allocation is used for the flows of products from plants to DCs and from RCs to plants, whereas the single-source allocation is implemented for the flows of products from the DCs to customers and from the customers to RCs. The single-source capacitated facility location problem (SSCFLP) can be considered as a simple form of the proposed SCLCND. According to Fisher *et al.* (1986), the SSCFLP is NP-hard itself and very hard to solve. To the best of our knowledge, there is no literature has implemented matheuristic as a solution methodology and this is the first time such an integrated bi-objective CLSC is thoroughly explored.

3. Problem description and mathematical modeling

An illustrative example of a CLSC is shown in Fig. 1. Here, the location of the plants and the customers are fixed and known, while the locations of the DCs and RCs are not. The chosen locations are selected from a list of potential candidate sites. In the forward SC, a set of plants manufacture a product delivered to selected DCs. A plant may supply the product to more than one DC. A product is then transferred to customers with the assumption that a customer is served by a single DC and its demand is deterministic. In the reverse SC, the used product is collected from the customers and transferred to the selected RCs. A customer is also assigned to one RC only. Some of the used products are recycled and shipped back to the plants, whereas the rest, known as non-salvageable products, is sent to the nearest disposal centre (landfill).

There are two types of costs, namely fixed and transportation costs. The former relates to the opening of DCs and RCs, which are based on the location and the capacity used. The latter depends on the type of vehicles used to transfer products. In this study, the environmental impact is measured by the amount of CO_2 emissions produced (). It is also assumed, as commonly adopted in the literature, that the amount of CO_2 emissions increases with the capacity of DCs or RCs, the distance traveled and the type of vehicle used. To ensure economies of scale, a minimal amount of products that can be shipped from a plant to a DC or from a RC to a plant is imposed. For cost efficiency, simplicity and convenience, each plant and each selected DC and RC use one type of transportation mode only, which is optimally chosen from a set of vehicles provided. This assumption increases the complexity of the model as a large number of binary variables are used. Practically, the use of low cost vehicles will reduce the transportation cost at the expense of a significant environmental impact. In other word, there is a trade-off between economic and environmental aspects when selecting the type of vehicle type. Therefore, the objective function that we set (minimizing total cost or amount of emissions) affects the solutions generated from the model.

The following notations are used to describe the sets and parameters of the proposed model.

TABLE 1 *A summary of previously relevant published literature for the CLSC*

No	Authors	Modelling	Aspect	Output	Solution Method
1	Fleischmann *et al.* (2001)	MILP	Eco	L, AL	Exact
2	Ko & Evans (2007)	MINLP	Eco	L, TA, FC	Genetic Algorithm
3	Lu & Bostel (2007)	MIP	Eco	L, AL	Lagrangian relaxation-based
4	Lee & Dong (2008)	MILP	Eco	L,TA	Tabu Search
5	Demirel & Gökçen (2008)	MILP	Eco	L, TA	Genetic Algorithm
6	Kusumastuti *et al.* (2008)	MILP	Eco	L, AL	Exact
7	Salema *et al.* (2009)	MILP	Eco	L, AL, TA, ND	Branch and Bound
8	Wang & Hsu (2010)	ILP	Eco	L, TA	Genetic Algorithm
9	Pishvaee & Torabi (2010)	MIP	Eco	L,FC,TA	Memetic Algorithm
10	Chaabane *et al.* (2012)	MILP	Eco, Env	L, TM	Exact
11	Pishvaee *et al.* (2010)	Fuzzy MIP	Eco, Env	L, TA	Fuzzy approach
12	Fahimnia *et al.* (2013)	MILP	Eco, Env	L, I, TA	Exact
13	Subramanian *et al.* (2013)	ILP	Eco	L, AL, TA	Simulated Annealing
14	Devika *et al.* (2014)	MILP	Eco, Env, Soc	L, AL, TA	Hybrid Metaheuristic
15	Ramezani *et al.* (2014)	Fuzzy MILP	Eco	L, AL, TM	Exact
16	Altmann & Bogaschewsky (2014)	LP	Eco, Env	L, AL, FC	Exact
17	Choudhary *et al.* (2015)	MILP	Eco, Env	L, AL, FC	Forest Data Structure
18	Kalaitzidou *et al.* (2015)	MILP	Eco	L, AL, FC	Branch and Bound
19	Chen *et al.* (2015)	MILP	Eco	L, AL, FC	Genetic Algorithm
20	Tiwari *et al.* (2016)	MILP	Eco, Env	L, AL	Hybrid Metaheuristic
21	Yi *et al.* (2016)	MILP	Eco	L, FC	Genetic Algorithm
22	Talaei *et al.* (2016)	MILP	Eco, Env	L, AL	ϵ-constraint
23	Kaya & Urek (2016)	MINLP	Eco	L, AL	Hybrid Metaheuristic
24	Kang *et al.* (2017)	Fuzzy	Eco, Env	L, AL	Particle Swarm
25	Govindan & Soleimani (2017)	MILP	Eco, Env	L, AL, TA, TM	Fuzzy Algorithm
26	Fathollahi-Fard *et al.* (2018)	MIP	Eco	L, AL, FC	Metaheuristic
27	Papen & Amin (2019)	MILP	Eco, Env	L, TA	ϵ-constraint

(*Continued*)

Table 1 *Continued*

No	Authors	Modelling	Aspect	Output	Solution Method
28	Patne *et al.* (2018)	MINLP	Eco	L, AL	Particle Swarm Optimisation
29	Li *et al.* (2018)	MINLP	Eco	L, AL	Hybrid Heuristic
30	Yadegari *et al.* (2019)	MILP	Eco	L, AL, FC	Hybrid Memetic Algorithm
31	Atabaki *et al.* (2019)	MILP	Eco	L, AL, FC, TA	Firefly Algorithm
32	Ghahremani-Nahr *et al.* (2019)	MINLP	Eco	L, AL, FC, TM	Whale Algorithm
33	Zhen *et al.* (2019)	MINLP	Eco	L, AL, FC	Tabu Search
34	Atabaki *et al.* (2020)	MILP	Eco, Env	L, AL, FC, TM	Robust Optimisation
35	Nayeri *et al.* (2020)	MIP	Eco, Env, Soc	L, AL, FC, TM	Goal Programming
36	Gholizadeh *et al.* (2021)	MILP	Eco, Env	L, AL, FC, TM	Robust and heuristic Optimisation
37	Salehi-Amiri *et al.* (2021)	MILP	Eco	L, AL, FC, TM	Exact, Metaheuristics, Hybrid Metaheuristics
38	**This paper**	MILP	Eco, Env	L, AL, FC, TM, ND	Matheuristic, Compromise Programming

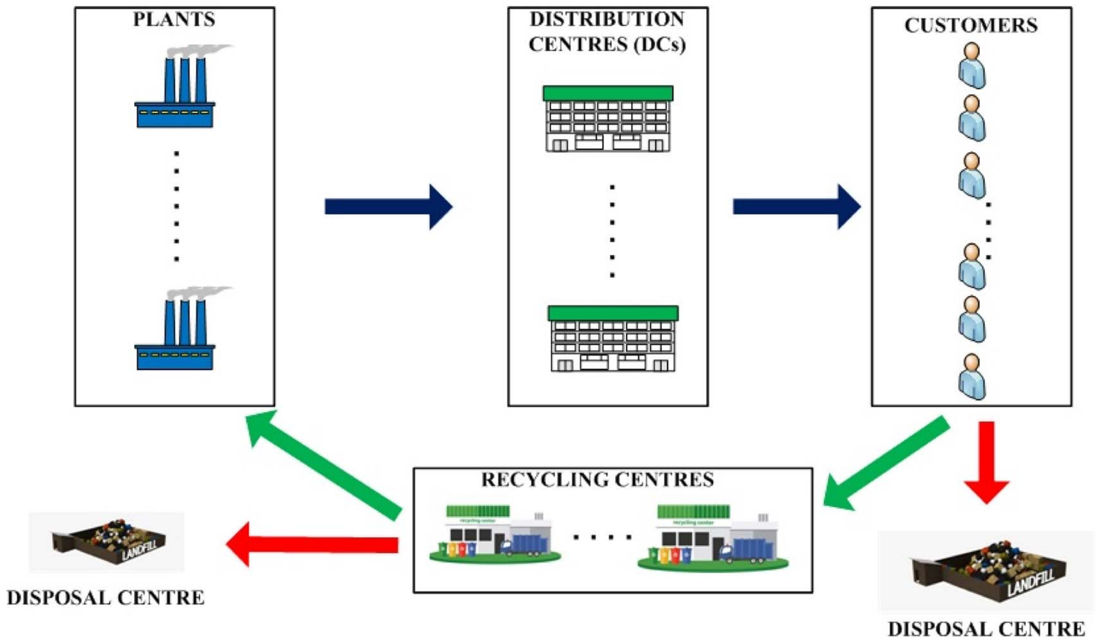


FIG. 1. Flow of products in the proposed CLSC problem.

Sets

I: set of plants with i as its index

J: set of potential DCs with j as its index

K: set of customers with k as its index

L: set of potential RCs with l as its index

H: set of distribution centre designs with h as its index

R: set of RC designs with r as its index

V^1: set of vehicles used to transport products from a plant to a DC with v as its index

V^2: set of vehicles used to transport products from a DC to a customer with v as its index

V^3: set of vehicles used to transport used products from a customer to a RC with v as its index

V^4: set of vehicles used to transport recycled materials from a RC to a plant with v as its index

Parameters

s_i: the capacity of plant $i \in I$

d_k: the demand of customer $k \in K$

$\bar{b}_h$: the number of products that can be stored in a potential DC when using design $h \in H$

$\ddot{b}_r$: the number of used products that can be recycled in a RC when using design $r \in R$

$\bar{f}_{jh}$: the fixed cost for opening DC $j \in J$ when using design $h \in H$

$\ddot{f}_{lr}$: the fixed cost for opening RC $l \in L$ when using design $r \in R$

$\bar{e}_h$: CO_2 emissions caused by opening a DC using design $h \in H$

$\ddot{e}_r$: CO_2 emissions caused by opening a RC using design $r \in R$

τ^1_{ijv}: the transportation cost to transfer one unit product from plant $i \in I$ to DC $j \in J$ using vehicle $v \in V^1$

τ^2_{jkv}: the transportation cost to transfer one unit product from DC $j \in J$ to customer $k \in K$ using vehicle $v \in V^2$

τ^3_{klv}: the transportation cost to transfer one unit used product from customer $k \in K$ to RC $l \in L$ using vehicle $v \in V^3$

τ^4_{liv}: the transportation cost to transfer one unit recycled material from RC $l \in L$to plant$i \in I$ using vehicle $v \in V^4$

ε^1_{ijv}: CO_2 emissions caused by transferring one unit product from plant $i \in I$ to DC $j \in J$ using vehicle $v \in V^1$

ε^2_{jkv}: CO_2 emissions caused by transferring one unit product from DC $j \in J$ to customer $k \in K$ using vehicle $v \in V^2$

ε^3_{klv}: CO_2 emissions caused by transferring one unit used product from customer $k \in K$ to RC $l \in L$ using vehicle $v \in V^3$

ε^4_{liv}: CO_2 emissions caused by transferring one unit recycled material from RC $l \in L$ to plant $i \in I$ using vehicle $v \in V^4$

ρ_i: the minimal amount of products transferred from plant $i \in I$ to a DC

θ_i: the minimal amount of recycled materials transferred from RC $l \in L$ to a plant

α: the average percentage of used products that can be collected from a customer to be recycled

β: the average percentage of used products that can be recycled and transformed into raw material.

This bi-objective closed-loop SC problem can be modelled as a mixed integer linear programming (MILP) as follows:

Decision variables

X^1_{ij}: the amount of products transported from plant $i \in I$ to DC $j \in J$

$$\bar{Y}_{jh} = \begin{cases} 1 & \text{if DC } j \in J \text{ uses design } h \in H, \\ 0 & \text{otherwise} \end{cases}$$

$$\ddot{Y}_{lr} = \begin{cases} 1 & \text{if RC } l \in L \text{ uses design } r \in R, \\ 0 & \text{otherwise} \end{cases}$$

$$U^1_{ijv} = \begin{cases} 1 & \text{if products are transferred from plant } i \in I \text{ to DC } j \in J \text{ using vehicle } v \in V^1, \\ 0 & \text{otherwise} \end{cases}$$

$$\hat{U}^1_{iv} = \begin{cases} 1 & \text{if plant } i \in I \text{ uses vehicle } v \in V^1 \text{ as its transportation mode,} \\ 0 & \text{otherwise} \end{cases}$$

$$U^2_{jkv} = \begin{cases} 1 & \text{if customer } k \in K \text{ is served by DC } j \in J \text{ using vehicle } v \in V^2, \\ 0 & \text{otherwise} \end{cases}$$

$$\hat{U}^2_{jv} = \begin{cases} 1 & \text{if DC } j \in J \text{ uses vehicle } v \in V^2 \text{ to transfer products to customers,} \\ 0 & \text{otherwise} \end{cases}$$

$$U^3_{klv} = \begin{cases} 1 & \text{if used products of customer } k \in K \text{ are shipped to RC } l \in L \text{ using vehicle } v \in V^3, \\ 0 & \text{otherwise} \end{cases}$$

$$\hat{U}^3_{lv} = \begin{cases} 1 & \text{if RC } l \in L \text{ uses vehicle } v \in V^3 \text{ to collect used products from customers,} \\ 0 & \text{otherwise} \end{cases}$$

$$U^4_{liv} = \begin{cases} 1 & \text{if recycled materials are transferred from RC } l \in L \text{ to plant} i \in I \\ & \text{using vehicle } v \in V^4, \\ 0 & \text{otherwise} \end{cases}$$

$$\hat{U}^4_{lv} = \begin{cases} 1 & \text{if RC } l \in L \text{ uses vehicle } v \in V^4 \text{ to transfer recycled materials to suppliers,} \\ 0 & \text{otherwise} \end{cases}$$

X^4_{li}: the amount of recycled materials transported from RC $l \in L$ to plant $i \in I$

Objective functions

$$\min Z_c = Z^{oc} + Z^{tc} \tag{3.1}$$

$$\min Z_e = Z^{oe} + Z^{te}, \tag{3.2}$$

where

$$Z^{oc} = \sum_{j \in J} \sum_{h \in H} \left(\bar{f}_{jh} \cdot \bar{Y}_{jh} \right) + \sum_{l \in R} \sum_{r \in R} \left(\ddot{f}_{lr} \cdot \ddot{Y}_{lr} \right) \tag{3.3}$$

$$Z^{tc} = \sum_{i\in I}\sum_{j\in J}\sum_{v\in V^1}\left(U^1_{ijv}\cdot X^1_{ij}\cdot \tau^1_{ijv}\right) + \sum_{j\in J}\sum_{k\in K}\sum_{v\in V^2}\left(U^2_{jkv}\cdot d_k\cdot \tau^2_{jkv}\right) + \sum_{k\in K}\sum_{l\in L}\sum_{v\in V^3}\left(U^3_{klv}\cdot \alpha\cdot d_k\cdot \tau^3_{klv}\right) + \sum_{l\in L}\sum_{i\in I}\sum_{v\in V^4}\left(U^4_{liv}\cdot X^4_{li}\cdot \tau^4_{liv}\right) \tag{3.4}$$

$$Z^{oe} = \sum_{j\in J}\sum_{h\in H}\left(\bar{e}_h\cdot \bar{Y}_{jh}\right) + \sum_{l\in R}\sum_{r\in R}\left(\ddot{e}_r\cdot \ddot{Y}_{lr}\right) \tag{3.5}$$

$$Z^{te} = \sum_{i\in I}\sum_{j\in J}\sum_{v\in V^1}\left(U^1_{ijv}\cdot X^1_{ij}\cdot \varepsilon^1_{ijv}\right) + \sum_{j\in J}\sum_{k\in K}\sum_{v\in V^2}\left(U^2_{jkv}\cdot d_k\cdot \varepsilon^2_{jkv}\right) + \sum_{k\in K}\sum_{l\in L}\sum_{v\in V^3}\left(U^3_{klv}\cdot \alpha\cdot d_k\cdot \varepsilon^3_{klv}\right) + \sum_{l\in L}\sum_{i\in I}\sum_{v\in V^4}\left(U^4_{liv}\cdot X^4_{li}\cdot \varepsilon^4_{liv}\right) \tag{3.6}$$

subject to

$$\sum_{j\in J} X^1_{ij} \le s_i, \forall i\in I \tag{3.7}$$

$$\sum_{i\in I} X^1_{ij} \le \sum_{h\in H}\left(\bar{Y}_{jh}\cdot \bar{b}_h\right),\ \forall j\in J \tag{3.8}$$

$$\sum_{h\in H} \bar{Y}_{jh} \le 1,\ \forall j\in J \tag{3.9}$$

$$X^1_{ij} \ge \rho_i\cdot \sum_{v\in V^1} U^1_{ijv},\ \forall i\in I, j\in J \tag{3.10}$$

$$X^1_{ij} \le M\cdot \sum_{v\in V^1} U^1_{ijv},\ \forall i\in I, j\in J \tag{3.11}$$

$$\hat{U}^1_{iv} \ge U^1_{ijv},\ \forall i\in I, j\in J, v\in V^1 \tag{3.12}$$

$$\sum_{v\in V^1} \hat{U}^1_{iv} \le 1,\ \forall i\in I \tag{3.13}$$

$$\sum_{k\in K}\sum_{v\in V^2}\left(U^2_{jkv}\cdot d_k\right)\leq\sum_{h\in H}\left(\bar{Y}_{jh}\cdot\bar{b}_h\right),\ \forall j\in J \tag{3.14}$$

$$\sum_{i\in I}X^1_{ij}=\sum_{k\in K}\sum_{v\in V^2}\left(U^2_{jkv}\cdot d_k\right),\ \forall j\in J \tag{3.15}$$

$$\sum_{j\in J}\sum_{v\in V^2}U^2_{jkv}=1,\ \forall k\in K \tag{3.16}$$

$$\hat{U}^2_{jv}\geq U^2_{jkv},\ \forall j\in J,k\in K,v\in V^2 \tag{3.17}$$

$$\sum_{v\in V^2}\hat{U}^2_{jv}\leq 1,\ \forall j\in J \tag{3.18}$$

$$\hat{U}^2_{jv}-\sum_{h\in H}\bar{Y}_{jh}\leq 0,\ \forall j\in J,v\in V^2 \tag{3.19}$$

$$\sum_{k\in K}\sum_{v\in V^3}\left(U^3_{klv}\cdot d_k\cdot\alpha\right)\leq\sum_{r\in R}\left(\ddot{Y}_{lr}\cdot\ddot{b}_r\right),\ \forall l\in L \tag{3.20}$$

$$\sum_{r\in R}\ddot{Y}_{lr}\leq 1,\ \forall l\in L \tag{3.21}$$

$$\sum_{l\in L}\sum_{v\in V^3}U^3_{klv}=1,\ \forall k\in K \tag{3.22}$$

$$\hat{U}^3_{lv}\geq U^3_{klv},\ \forall k\in K,l\in L,v\in V^3 \tag{3.23}$$

$$\sum_{v\in V^3}\hat{U}^3_{lv}\leq 1,\ \forall l\in L \tag{3.24}$$

$$\hat{U}^3_{lv} - \sum_{r \in R} \ddot{Y}_{lr} \leq 0, \ \forall l \in L, v \in V^3 \tag{3.25}$$

$$\sum_{k \in K} \sum_{v \in V^3} \left(U^3_{klv} \cdot d_k \cdot \alpha \cdot \beta \right) \leq \sum_{i \in I} X^4_{li}, \ \forall l \in L \tag{3.26}$$

$$\sum_{l \in L} X^4_{li} \leq \sum_{j \in J} X^1_{ij}, \ \forall i \in I \tag{3.27}$$

$$X^4_{li} \geq \theta_l \cdot \sum_{v \in V^4} U^4_{liv}, \ \forall l \in L, i \in I \tag{3.28}$$

$$X^4_{li} \leq M \cdot \sum_{v \in V^4} U^4_{liv}, \ \forall l \in L, i \in I \tag{3.29}$$

$$\hat{U}^4_{lv} \geq U^4_{liv}, \ \forall l \in L, i \in I, v \in V^4 \tag{3.30}$$

$$\sum_{v \in V^4} \hat{U}^4_{lv} \leq 1, \ \forall l \in L \tag{3.31}$$

$$\hat{U}^4_{lv} - \sum_{r \in R} \ddot{Y}_{lr} \leq 0, \ \forall l \in L, v \in V^4 \tag{3.32}$$

$$\bar{Y}_{jh} \in \{0, 1\}, \ \forall j \in J, h \in H \tag{3.33}$$

$$\ddot{Y}_{lr} \in \{0, 1\}, \ \forall l \in L, r \in R \tag{3.34}$$

$$X^1_{ij} \geq 0, \text{integer}, \ \forall i \in I, j \in J \tag{3.35}$$

$$U^1_{ijv} \in \{0,1\},\ \forall i \in i, j \in J, v \in V^1 \tag{3.36}$$

$$\hat{U}^1_{iv} \in \{0,1\},\ \forall i \in I, v \in V^1 \tag{3.37}$$

$$U^2_{jkv} \in \{0,1\},\ \forall j \in J, k \in K, v \in V^2 \tag{3.38}$$

$$\hat{U}^2_{jv} \in \{0,1\},\ \forall j \in J, v \in V^2 \tag{3.39}$$

$$U^3_{klv} \in \{0,1\},\ \forall k \in K, l \in L, v \in V^3 \tag{3.40}$$

$$\hat{U}^3_{lv} \in \{0,1\},\ \forall l \in L, v \in V^3 \tag{3.41}$$

$$U^4_{liv} \in \{0,1\},\ \forall l \in L, i \in I, v \in V^4 \tag{3.42}$$

$$\hat{U}^4_{lv} \in \{0,1\},\ \forall l \in L, v \in V^4 \tag{3.43}$$

$$X^4_{li} \in \{0,1\},\ \forall l \in L, i \in I, \tag{3.44}$$

where M is a very large positive number.

Objectives (3.1) and (3.2) refer to the economic and environmental impacts, respectively. Both objectives consist of two parts, namely, the sum of all fixed costs/emissions and the sum of all transportation costs/emissions. The first term of the total transportation costs/emissions formulation (Z^{tc} and Z^{te}) in (3.4) and (3.6) makes the problem nonlinear due to the product of the two decision variables (U^1_{ijv} and X^1_{ij}). In the implementation, the problem is transformed into a linear problem (MILP) by the introduction of new decision variables and constraints.

Constraints (3.7) and (3.8) are the capacity constraints, where a plant and an open DC cannot, respectively, deliver and receive products more than their capacities. Constraint (3.9) enforces that one design (with a certain capacity) can only be used by an open DC. Constraints (3.10)–(3.11) express that the amount of products shipped to a DC from a plant must be more than or equal to the lower limit.

Constraints (3.12)–(3.13) guarantee that a plant only uses one type of vehicle for transporting products to DCs. Equation (3.14) indicates the maximum amount of products; an open DC can deliver to its customers (i.e. up to its capacity). Constraint (3.15) states the flow conservation constraints through an open DC. A single-source allocation problem is represented by Constraint (3.16), where a customer is only served by one open DC. This constraint is widely used in the facility location problem. Constraints (3.17)–(3.18) ensure that an open DC only uses one type of vehicle to deliver products to its customers, whereas Constraint (3.19) imposes that a vehicle for transporting products is only required by an open DC.

The interpretation of Constraints (3.20) and (3.21) is, respectively, similar to Constraints (3.8) and (3.9), with DCs replaced by RCs. Constraint (3.22) assures that the used products of a customer are only collected by one open RC. Constraints (3.23)–(3.24) define that an open RC only uses one type of vehicle to collect used products from its customers. Constraint (3.25) enforces that a vehicle for collecting used products is only needed by an open RC. The flow conservation constraints through an open RC are represented by Constraint (3.26). Constraint (3.27) ensures that the amount of recycled products delivered to a plant does not exceed the plant capacity. Equations 3.283.29 impose that the amount of recycled products transferred to a plant from RC must be more than or equal to the lower limit. Constraints (3.30)–(3.31) ensure that an open RC only uses one type of vehicle to ship recycled products to the plants. Constraint (3.32) conveys that a vehicle for transferring recycled products to the plants is only needed by an open RC. Equations 3.333.44 indicate binary and non-negativity restrictions on decision variables.

Constraints (3.7)–(3.8) and (3.20) are designed based on a well-known capacitated plant location problem (Sridharan, 1995). Constraints (3.9) and (3.21) have been applied for the location problems with multiple capacity levels (Correia *et al.*, 2010). Constraints (3.15) and (3.26) are usually used by the two-stage capacitated facility location model (Irawan *et al.*, 2016).

4. Compromise programming for the bi-objective CLSC

In real-life, we may face problems with more than one objective, which is referred to as a multi-objective problem (Kaveh *et al.*, 2019; Song *et al.*, 2020; Yakavenka *et al.*, 2019). There are several methods that can be used to deal with multi-objective problems including goal programming, Pareto efficient set generation and CP. In this paper, CP is chosen as it has the advantage of not requiring the goal target values information, as in goal programming, while at the same time being relatively faster than the Pareto efficient set generation technique. The idea of CP is to select a solution from the set of efficient solutions based on the assumption that any decision-maker seeks a solution as close to the ideal point as possible (Romero & Rehman, 1989). According to Jones (2011), CP minimises a set of weighted, scaled distances between the ideal and efficient solutions. Recently, CP has been used successfully in many applications, including in the environmental area of wind farm scheduling as shown by Irawan *et al.* (2017a) and references therein.

CP uses a distance function to measure the closeness between a solution and the ideal point, where a family of L_p metrics is usually implemented. The general formulation of a CP approach is expressed as follows:

$$\min L_p = \left(\sum_{k=1}^{N} \left| \hat{w}_k \cdot \frac{Z_k(x) - Z_k^*}{Z_k^+ - Z_k^*} \right|^p \right)^{\frac{1}{p}}, \tag{4.1}$$

where

p: indicates the distance measure with p in the range $[1, \infty]$,

N: the number of objectives,

Z_k^*: the ideal solution of objective o,

Z_k^+: the anti-ideal solution of objective o,

$Z_k(x)$: the compromise solution that minimises L_p, and

$\hat{w}_k$: the weight/importance of objective o relative to the other objectives.

In this study, we explore the case when the value of p is 1 and ∞.

For $p = 1$, Equation (4.1) is transformed into the following form:

$$\min L_1 = \min \sum_{k=1}^{N} \hat{w}_k \cdot \frac{Z_k(x) - Z_k^*}{Z_k^+ - Z_k^*}. \tag{4.2}$$

When $p = \infty$, the objective function (4.1) reduces to minimizing the maximum deviation (π) as follows:

$$\min L_\infty = \min \pi \tag{4.3}$$

$$\text{s.t}\hat{w}_k \cdot \frac{Z_k(x) - Z_k^*}{Z_k^+ - Z_o^*} \leq \pi, \forall k = 1, \ldots, N. \tag{4.4}$$

Algorithm 1 shows the main steps of CP for solving the bi-objective CLSC network problem. This approach consists of three phases. The first phase is to obtain the anti-ideal solution for each objective, where the maximising problem is used instead. As expected, in the optimal solution configuration, all potential sites with their largest capacity will be chosen for opening DCs and RCs. Moreover, a customer will be served by the furthest facility instead of the nearest one. This property of the maximizing problem renders its resolution relatively easy to solve optimally. In the second phase, the ideal solution for each objective is obtained by solving the minimizing problem separately. In this study, even though the problem is more complex, it is also solved by an exact method, where CPLEX is used to generate optimal or near optimal solutions.

Phase 3 is introduced to find the solutions that minimize L_1 and L_∞ as these can serve as bounds for the compromise solutions. The MILPs for L_1 and L_∞ are solved by both the exact method using CPLEX and the proposed matheuristic. For relatively large instances, the matheuristic is an effective tool for minimizing L_1 and L_∞ problems as these problems are relatively harder to solve optimally.

Algorithm 4.1 The main phases of the CP procedure

Phase 1

Solve optimally the maximising total cost problem (Equation 3.1) subject to constraints (3.7) to (3.44) and let Z_c^+ be the anti-ideal total cost.

Solve optimally the maximising total emissions problem (Equation 3.2) subject to constraints (3.7) to (3.44) and let Z_e^+ be the anti-ideal total amount of emissions.

Phase 2

Solve optimally the minimising total cost problem (Equation 3.1) subject to constraints (3.7) to (3.44) and let Z_c^* be the ideal total cost.

Solve optimally the minimising total emissions problem (Equation 3.2) subject to constraints (3.7) to (3.44) and let Z_e^* be the ideal total amount of emissions.

Phase 3

Using the exact method or the proposed matheuristic, solve the minimising L_1 problem subject to constraints (3.7) to (3.44) where

$$L_1 = \frac{\tilde{w} \cdot (Z_c - Z_c^*)}{Z_c^+ - Z_c^*} + \frac{(1-\tilde{w}) \cdot (Z_e - Z_e^*)}{Z_e^+ - Z_e^*} \tag{4.5}$$

and $\tilde{w}$ is the weight (parameter) of the first objective (total cost).
Let Z_c^1 denote the total cost obtained and Z_e^1 the total amount of emissions.

Using the exact method or the proposed matheuristic, solve the minimising L_∞ problem where

$$L_\infty = \pi \tag{4.6}$$

subject to constraints (3.7) to (3.44) with the following additional constraints:

$$\frac{\tilde{w} \cdot (Z_c - Z_c^*)}{Z_c^+ - Z_c^*} \leqslant \pi \tag{4.7}$$

$$\frac{(1-\tilde{w}) \cdot (Z_e - Z_e^*)}{Z_e^+ - Z_e^*} \leqslant \pi \tag{4.8}$$

Let Z_∞^1 be the total cost obtained and Z_e^∞ the total amount of emissions.

Compromise solutions are those solutions bounded by L_1 and L_∞.

5. The proposed matheurisic

A matheuristic technique, which falls within the class of hybridisation of heuristics and exact methods, is proposed. In this study, this approach is designed by integrating an aggregation technique, an exact method, a local search and metaheuristic. For the applications of matheuristics, the readers are referred to Ramos *et al.* (2020), Huber *et al.* (2020), Obal *et al.* (2019) and Irawan & Jones (2019). This type of approach, which was shown to be efficient for solving a class of location problems (Irawan *et al.*, 2017b), is adapted to tackle the L_1 and L_∞ problems.

5.1 *Overview of the algorithm*

This method requires the ideal and anti-ideal solutions of the total costs and the amount of emissions that have been calculated in Phases 1 and 2 of Algorithm 4.1. Algorithm 5.1 presents the proposed matheuristic which consists of five stages.

Algorithm 5.1 The proposed matheuristic approach

Define T, $\bar{\mu}$, $\ddot{\mu}$, τ, τ' and τ''.
Set $\underline{\bar{Y}}_{jh}=0,\forall j\in J,h\in H$; $\underline{\bar{Y}}_{lr}=0,\forall l\in L,r\in R$; $\underline{\hat{U}}^1_{iv}=0,\forall i\in I,v\in V^1$; $\underline{\hat{U}}^2_{jv}=0,\forall j\in J,v\in V^2$; $\underline{\hat{U}}^3_{lv}=0,\forall l\in L,v\in V^3$; $\underline{\hat{U}}^4_{lv}=0,\forall l\in L,v\in V^4$; and $Z=\infty$.
Stage 1 (forward supply chain)
  repeat
    Aggregate $|J|$ to $\bar{\mu}$ potential DC sites using a random approach.
    Solve the aggregated L_1 or L_∞ problems using the exact method (CPLEX) within (τ) seconds.
    Update $\underline{\hat{U}}^1_{iv}=1,\forall i\in I,v\in V^1$, $\underline{\bar{Y}}_{jh}=1,\forall j\in J,h\in H$ and $\underline{\hat{U}}^2_{jv}=1,\forall j\in J,v\in V^2$ if they are in the solution of aggregated problem.
  until T times
Stage 2 (reverse supply chain)
  repeat
    Aggregate $|L|$ to $\ddot{\mu}$ potential RC sites using a random approach.
    Solve the aggregated L_1 or L_∞ problems using the exact method (CPLEX) within (τ) seconds.
    Update $\underline{\ddot{Y}}_{lr}=1,\forall l\in L,r\in R$, $\hat{U}^3_{lv}=1,\forall l\in L,v\in V^3$ and $\hat{U}^4_{lv}=1,\forall l\in L,v\in V^4$ if they are in the solution of aggregated problem.
  until T times
Stage 3 (closed-loop supply chain)
  Solve the original (without aggregation) minimising L_1 or L_∞ problems using the exact method (CPLEX) within (τ) seconds. Here, the binary decision variables are set to 0 if they are not selected when solving the aggregated problems in the previous stages.
  Let Z be its objective function with all the decision variables also obtained.
Stage 4 (interchange-based heuristic or VNS)
  Implement either the proposed interchange-based heuristic or VNS for the forward and reverse supply chain
Stage 5 (Finalisation)
  Solve the original L_1 or L_∞ problems using the exact method (CPLEX) within (τ') seconds. Here, the obtained values from the previous stage for $\hat{U}^1_{iv}$, $\bar{Y}_{jh}$, $\hat{U}^2_{jv}$, $\hat{U}^3_{lv}$ and $\hat{U}^4_{lv}$ are known. The solution includes the flows of products from plants to customers and from customers to plants. Based on the solution obtained, the total cost (Z_c) and the total emissions (Z_e) are calculated.

For the initialisation stage of Algorithm 5.1, the necessary parameters are defined. This includes the number of iterations (T) to solve the aggregated problems, the number of aggregated potential DCs $(\bar{\mu})$ and RCs $(\ddot{\mu})$, the maximum computational time for CPLEX to solve the aggregated problems (τ), the augmented problem (τ') and the reduced problem (τ''). A set of arrays is also constructed to store the solutions obtained when solving the aggregated problems. The data structure of these arrays is the same as the one representing the decision variables in the model.

Stages 1 and 2 use an aggregation technique to solve the problems where an iterative process is conducted. In these stages, the aim is to generate promising sites to locate DCs and RCs using an aggregation approach. Given that the CLSC problem can be divided into forward SC and reverse SC problems, the resulting problems become relatively easier to solve. The ideal and anti-ideal solutions of the total cost and amount of emissions for both SCs are calculated based on the solutions generated in Phases 1 and 2 of Algorithm 4.1.

In the first stage, a set of aggregation problems on the forward SC is generated. Firstly, $\bar{\mu}$ potential DCs are selected randomly out of $|J|$ sites. The aggregated forward SC problem (minimizing L_1 or L_∞) consisting of $|I|$ plants, $(\bar{\mu})$ (instead of $|J|$) potential DCs and $|K|$ customers is then solved by CPLEX within (τ) seconds. For the forward SC problem, model (3.1)–(3.44) is reduced by considering the following decision variables only: $\bar{Y}_{jh}$, X^1_{ij}, U^1_{ijv}, $\hat{U}^1_{iv}$, U^2_{jkv} and $\hat{U}^2_{jv}$. Note that the objective function formulation needs to be revised by removing the flows of products from customers to plants and the opening of RCs. Constraints (3.20)–(3.32) are also not needed here. The obtained solution is stored in arrays $\underline{\hat{U}}^1_{iv}$, $\underline{\bar{Y}}_{jh}$ and $\underline{\hat{U}}^2_{jv}$ for decision variables $\hat{U}^1_{iv}$, $\bar{Y}_{jh}$ and $\hat{U}^2_{jv}$, respectively. The process is repeated T times.

The main procedure of Stage 2 is relatively similar to that of Stage 1 whereby a set of aggregation problems is constructed on the reverse SC instead. The potential RCs are also aggregated to $\ddot{\mu}$ sites randomly selected from $|L|$ sites. The aggregated reverse SC problem comprises $|K|$ customers, $\ddot{\mu}$ potential RCs and $|I|$ plants which is solved by CPLEX within (τ) seconds. Note that this is a reduced model which includes the following decision variables only: $\ddot{Y}_{lr}$, X^4_{li}, U^3_{klv}, $\ddot{U}^3_{lv}$, U^4_{liv} and $\ddot{U}^4_{lv}$. Constraints (3.7)–(3.19) are also excluded from the model. In addition, in constraints (3.27), $\sum_{j\in J} X^1_{ij}$ is approximated by s_i (capacity of plant i) as the variable X^1_{ij} belongs to the forward SC. The obtained solution is stored in arrays $\underline{\ddot{U}}^3_{lv}$, $\underline{\ddot{Y}}_{lr}$ and $\underline{\ddot{U}}^4_{lv}$ for decision variables $\ddot{U}^3_{lv}$, $\ddot{Y}_{lr}$ and $\ddot{U}^4_{lv}$, respectively.

A feasible solution for the original problem can be found easily. This is then used as an initial solution in the next stage, which is the local search. Let Z be its objective function. The values of all the corresponding decision variables are then copied into the set of arrays ($\underline{\bar{Y}}_{jh}$, $\underline{\ddot{Y}}_{lr}$, $\underline{X}^1_{ij}$, $\underline{U}^1_{ijv}$, $\underline{\hat{U}}^1_{iv}$, $\underline{U}^2_{jkv}$, $\underline{\hat{U}}^2_{jv}$, $\underline{U}^3_{klv}$, $\underline{\hat{U}}^3_{lv}$, $\underline{U}^4_{liv}$, $\underline{\hat{U}}^4_{lv}$ and $\underline{X}^4_{li}$) which are referred to as the best storage arrays. Let χ and ψ denote the set of open DCs and RCs, respectively.

In Stage 4, we propose two methods to improve the quality of the solutions produced by the previous stages, namely the interchange-based heuristic and a metaheuristic based on Variable Neighbourhood Search (VNS). The description of these two methods is presented in the next subsections. Here, the implementation of Matheuristic with the interchange-based heuristic is called MTH-ICH, whereas the one with the VNS is refer to as MTH-VNS. In the final stage, the original L_1 or L_∞ problem is then solved using the exact method with CPLEX within (τ'') seconds. This problem can be considered as the reduced problem given that the binary decision variables $\hat{U}^1_{iv}$, $\bar{Y}_{jh}$, $\hat{U}^2_{jv}$, $\hat{U}^3_{lv}$ and $\hat{U}^4_{lv}$ are now treated as known. These are populated from the storage arrays obtained from the previous stage, namely, the local search. Here, the flows of products from plants to customers and the flows of returned products from customers to plants are obtained. Based on the solution found, the total cost (Z_c) and the total amount of emissions (Z_e) are then determined.

5.2 *The interchange-based heuristic (Stage 4 of Algorithm 5.1)*

The interchange heuristic is developed using a combination of the first and best improvement strategy. The heuristic is divided into two categories, namely, one for the forward SC and another for the reverse SC. The former local search seeks the best location of open DCs, whereas the latter searches for the RCs' sites. The algorithms for both supply chains are quite similar.

Heuristic for the forward SC

The main steps of the proposed heuristic for the forward SC are presented in Algorithm 5.2. The algorithm aims to seek a potential DC site to replace a DC site already used in the current solution.

Algorithm 5.2 The proposed heuristic for the forward supply chain

---

Require: Incumbent solution χ with its objective function value Z

1: Define cpu_{max}, (τ'''), $\hat{\tau}$ and ρ.
2: **for** each open DC $\hat{j} \in \chi$ (current solution) **do**
3:     **If** $CPU > cpu_{max}$ **then** break.
4:     Determine the set ς as the list of ρ potential DCs nearest to open DC $\hat{j}$ where $\varsigma \not\subset \chi, \varsigma \subseteq J$.
5:     Set $\lambda = 0$ (Best Saving).
6:     **for** each potential DC $\tilde{j} \in \varsigma$ **do**
7:         Construct new storage arrays ($\underline{\bar{Y}'}_{jh}$, $\underline{\ddot{Y}'}_{lr}$, $\underline{\hat{U}'}^{1}_{iv}$, $\underline{\hat{U}'}^{2}_{jv}$, $\underline{U}'^{3}_{klv}$, $\underline{\hat{U}'}^{3}_{lv}$, $\underline{U}'^{4}_{liv}$, $\underline{\hat{U}'}^{4}_{lv}$ and $\underline{X}'^{4}_{li}$). Copy the best storage arrays values into these arrays
8:         Update $\underline{\bar{Y}'}_{\tilde{j}h} \leftarrow \underline{\bar{Y}}_{\hat{j}h}, \underline{\bar{Y}'}_{\hat{j}h} = 0, \forall h \in H$ and $\underline{\hat{U}'}^{2}_{\tilde{j}v} \leftarrow \underline{\hat{U}}^{2}_{\hat{j}v}, \underline{\hat{U}'}^{2}_{\hat{j}v} = 0, \forall v \in V^2$
9:         Treat all decisions variables as known except X^1_{ij}, U^1_{ijv} and U^2_{jkv}.
10:         Solve the problem using CPLEX within (τ''') seconds. Let Z' (for L_1 or L_∞ problems) be its objective function value.
11:         **If** $(Z - Z') > \lambda$ **then** set $\lambda = Z - Z'$ and $\vec{j} = \tilde{j}$
12:     **end for**
13:     **if** $\lambda > 0$ **then**
14:         Construct new storage arrays ($\underline{\bar{Y}'}_{jh}$, $\underline{\ddot{Y}'}_{lr}$, $\underline{\hat{U}'}^{1}_{iv}$, $\underline{\hat{U}'}^{2}_{jv}$, $\underline{\hat{U}'}^{3}_{lv}$ and $\underline{\hat{U}'}^{4}_{lv}$. Copy the best storage arrays values into these arrays
15:         Update $\underline{\bar{Y}'}_{\vec{j}h} \leftarrow \underline{\bar{Y}}_{\hat{j}h}, \underline{\bar{Y}'}_{\hat{j}h} = 0, \forall h \in H$ and $\underline{\hat{U}'}^{2}_{\vec{j}v} \leftarrow \underline{\hat{U}}^{2}_{\hat{j}v}, \underline{\hat{U}'}^{2}_{\hat{j}v} = 0, \forall v \in V^2$
16:         Treat all decisions variables as known except X^1_{ij}, U^1_{ijv}, U^2_{jkv}, U^3_{klv}, X^4_{li} and U^4_{liv}.
17:         Solve the problem using CPLEX within $\hat{\tau}$ seconds. Let Z' (for L_1 or L_∞ problems) be its objective function value.
18:         If $(Z - Z') > 0$ then set $Z = Z'$ and update all best storage arrays based on the decision variables values obtained and go back to Line 3.
19:     **end if**
20: **end for**
21: Return Z and all the best storage arrays.

Firstly, the maximum CPU time ($cpu_{\max}$) and the number of nearest potential DCs (ϱ) from an open DC are defined. In Lines 2–20, an open DC, say DC $\hat{j}$, is swapped with each potential DC site included in the set ς (the list of (ϱ) potential DCs nearest to the open DC $\hat{j}$, where $\varsigma \not\subset \chi$, $\varsigma \subseteq J$). The set ς is introduced to reduce the computational time at the expense of a small quality loss. For more information on the design and practicality of neighborhood reduction, see Salhi (2017). Here, the open DC is not necessarily swapped with the potential DC located too far from the open DC. The potential DC site that yields the best positive saving is chosen to replace DC $\hat{j}$. Then, the process returns to Line 2 without checking the remaining open DCs that have not been searched.

In Lines 6–12, the chosen DC is restricted to opt for the same design (capacity) and to use the same type of vehicle already present at the removed DC. The model is reduced to find the flow of products from plants to customers only (forward SC). In other words, all the decision variables except X^1_{ij}, U^1_{ijv} and U^2_{jkv} are treated as known, which are populated from the best storage arrays. The reduced problem is then solved by the exact method using CPLEX applied within (τ''') seconds. In Lines 13–19, once the potential DC that produces the best improvement is found, the flows of new and returned products are determined by solving the problem which considers the following decision variables X^1_{ij}, U^1_{ijv}, U^2_{jkv}, X^4_{li}, U^3_{klv} and U^4_{liv} only. At this stage, other binary decision variables have known values which are populated by the best storage arrays. This reduced problem is solved by CPLEX within $\hat{\tau}$ seconds. The search goes back to Line 2 and the process is repeated until there is either no improvement or the computational time reached $cpu_{\max}$, whichever comes first.

Heuristic for reverse supply chain

The local search for the reverse SC is relatively similar to the one in the forward SC given in Algorithm 3, except that

(i) the operator, which finds the location of the DC site, seeks a potential RC site to replace an RC site already used in the current solution,

(ii) in Lines 6–12, the model is also reduced to determining the flow of returned products from customers to plants (reverse SC). Here, all decision variables, except X^4_{li}, U^3_{klv} and U^4_{liv}, are known and are populated from the best storage arrays,

(iii) in Lines 13–19, the flows of new and returned products are determined by solving the reduced problem, which considers the following decision variables X^1_{ij}, U^1_{ijv}, U^2_{jkv}, X^4_{li}, U^3_{klv} and U^4_{liv} only.

5.3 *The proposed variable neighbourhood search (Stage 4 of Algorithm 5.1)*

VNS was first formally formulated by Hansen & Mladenović (1997) for solving the p-median problem. VNS consists of two parts, namely neighbourhood search and local search where the objective of the first part is to help the search process escape from the local optima. The local search seeks the best solution in the local neighbourhood. A larger neighbourhood is used if the local search process cannot find any improvement, otherwise it reverts to the smaller neighbourhood. VNS-based matheuristic has successfully been implemented to address challenging problems, including for location problem (Irawan *et al.*, 2017c), vehicle routing problem (Wang *et al.*, 2017) and layout problem (Irawan *et al.*, 2019).

The same as the interchange heuristic described previously, the proposed VNS aims to address the forward and reverse SCs. Algorithm 4 presents the main steps of the proposed VNS where parameter $k_{\max}$ needs to be defined first.

In the proposed, the shaking process is conducted by removing a facility randomly selected from the current solution (χ') and replacing it with a randomly selected potential site near to the removed facility. Note that the site is chosen from a set of ς, a list of potential DCs near to removed DC, which is described in the previous subsection. Once the interchange has been conducted, the allocation problem is solved and the objective value z' is calculated. Here, the allocation problem is the same as the one presented in Line 10 of Algorithm 3 which is solved by an exact method. This shaking process is repeated k times to perturb the solution.

Then, an interchange heuristic is proposed using the best improvement strategy to improve the quality of solution by finding the local optima. The algorithm aims to seek the best facility location site to be swapped with the facility site used in the current solution. To speed up the process, the swap is performed between facility $j \in \chi'$ and a potential site ($\hat{j} \in \varsigma, \hat{j} \notin \chi'$) which is near to facility j. The restricted allocation problem is solved to check whether improvement has been made. Note that the allocation problem is restricted by only including facility $\hat{j}$ and a set of open facilities near to facility $\hat{j}$. This significantly reduces the computing time at the expense of a relatively small solution quality reduction. The permanent swap between the best potential site and the best facility to be removed will be done if improvement occurs. The local search process will be repeated until no improvement is found.

In Move or Not step, if the proposed heuristic is not able to improve the solution, a larger neighbourhood is systematically used otherwise the smallest one will be used. This can be performed by updating the value of k, where $k = k_{\max}$ indicates the largest neighbourhood, while $k = 1$ represents the smallest one. In the VNS, the smallest neighbourhood is the one that is closest to the current solution, whereas the largest one is the farthest from the current solution (Hansen & Mladenović, 1997). Similar to the previous method, the proposed VNS can also be used for the reverse SC. The modification of the algorithm is quite similar with the previous method. Here, DC sites are replaced by RC sites and all decisions for the flow of forward SC are replaced by the ones for reverse SC.

Algorithm 5.3 The proposed VNS for the forward supply chain

Require: Incumbent solution χ with its objective function value z
Define $k_{\max}$ and set $\chi' \leftarrow \chi$
Set $k = 1$
Perform Shaking Procedure as follows:
  for i=1 to k **do**
    Choose randomly an open DC in current solution, say DC $j \in \chi'$
    Pick randomly a potential site near to facility j, say facility $\hat{j}$, $\hat{j} \in J, \hat{j} \notin \chi'$
    In current solution χ', replace facility j with $\hat{j}$
    Solve the allocation problem using solution χ' and determine objective value z'
  end for
Execute the interchange heuristic using the following steps:
  Set improve = True
  while improve **do**
    Update improve = False and Set $z_b = \infty$
    for each $j \in \chi'$ **do**
      for each $\hat{j} \in J, \hat{j} \notin \chi'$ **do**
        if site $\hat{j}$ is near to facility j **then**
          Update $\chi'' \leftarrow \chi'$
          In solution χ'', replace facility j with $\hat{j}$
          Solve the restricted allocation problem using solution χ'' and determine objective value z''
          if $z'' < z_b$ **then** Update $z_b = z''$, $j_b = j$ and $\hat{j}_b = \hat{j}$
        end if
      end for
    end for
    if $z_b < z'$ **then**
      Update $z' = z_b$, replace facility j_b with $\hat{j}_b$ in solution χ'
      Update improve = True
    end if
  end while
Move or Not:
  if $z' < z$ **then** Update $z = z'$, $\chi \leftarrow \chi'$ and $k = 1$
  else Update $z' = z$, $\chi' \leftarrow \chi$ and $k = k + 1$
return z and χ

6. Computational experiments

Computational experiments are carried out to examine the performance of the proposed solution method. The implementation is written in C++. Net 2015 and the mathematical model is solved using the IBM

ILOG CPLEX version 12.7 Concert Library. The tests are run on a PC with an Intel Core i7 CPU @ 3.60GHz processor, 16.00 GB of RAM and under Windows 7. To the best of our knowledge, no benchmark dataset is available for this problem. We constructed four new datasets with $|K| = 50 - 200$ with an increment of 50.

6.1 *Computational evaluation of the proposed matheuristic*

The four newly constructed datasets are used to assess the performance of the proposed solution method. For each instance, the number of potential DCs and RCs is set to $|K|$ (i.e. $|J| = |L| = |K|$), with the number of plants ($|I|$) being set to $0, 1, \ldots, |K|$. The locations of plants, potential DCs, customers and potential RCs are randomly and uniformly generated. The demand of each customer is randomly chosen between 5 and 15, whereas the number of designs for DCs and RCs is set to 3 (i.e. $|H| = |R| = 3$). The number of vehicle types for plants, DCs and RCs is also set to 3 (i.e. $|V^1| = |V^2| = |V^3| = |V^4| = 3$). The production capacity, DC and RC capacity for each design, along with their associated parameters, are estimated based on the total demand of customers. Here, the dataset is constructed in such a way that the total transportation cost/emissions obtained is close to the total fixed/opening cost/emissions. These datasets can be downloaded from .

Table 2 shows the ideal and anti-ideal solutions found by CPLEX when solving the minimizing/-maximizing total cost and emissions problems. In the experiments, the computing time of CPLEX is limited to 3 h where the upper bound (UB) and the lower bound (LB) are obtained. Therefore, the Gap (%) is determined using the following equation:

$$Gap(\%) = \frac{\text{UB} - \text{LB}}{\text{UB}} \times 100. \tag{6.1}$$

For the minimizing problems, the required number of DCs (p) and the number of RCs (q) to be opened are also provided. Table 2 reveals that the maximizing problems can be easily solved by CPLEX. For the minimizing problems, CPLEX produced near-optimal solutions; on average, relatively small gaps of 1.2 and 1.86% for the minimizing total cost and emissions problems, respectively.

The proposed matheuristic is used for the minimizing L_1 and L_∞ problems, where the weight of objectives is set equally to 0.5. The solutions for L_1 and L_∞ problems found by CPLEX are used for comparison purposes. Here, the UB found within the maximum allowed time of 3 h is used as the objective function value. According to the results, CPLEX was not able to obtain the LB within the allowed time. Therefore, the performance of the matheuristic is then measured using the percentage deviation (Dev) instead of Gap (Equation 6.1) where Dev (%) is computed as follows:

$$Dev(\%) = \frac{Z' - Z^b}{Z^b} \times 100, \tag{6.2}$$

where Z' refers to the objective function value obtained by either the exact method (UB) or the proposed method, whereas Z^b is the best objective function value attained by either the exact method or the proposed matheuristic.

In these experiments, the following parameter values are used: $T = 2$, $\tau = 2|K|/5$, $\tau' = 2|K|$, $\tau'' = 50$, $\tau''' = 1$, $\hat{\tau} = 5$, $cpu_{\max} = 2|K|$, $\varrho = 25$ and $k_{\max} = 2$. Parameters τ, τ', τ'', τ''', $\hat{\tau}$, $cpu_{\max}$ are measured in seconds. The number of aggregated DCs ($\bar{\mu}$) and RCs ($\ddot{\mu}$) is calculated based on the UB

Table 2 *The summary results of minimizing/maximizing total cost and emissions problem*

|K|	Exact Method on the minimizing/maximizing cost problems									
	Minimizing Z_c							Maximizing Z_c		
	UB	Gap(%)	Z^{oc} (%)	Z^{tc} (%)	p	q	CPU(s)	UB	Gap(%)	CPU(s)
50	180,302	0.75	59.26	40.74	6	5	10,804	1,777,546	0.00	1.40
100	543,406	0.14	54.47	45.53	8	5	10,800	8,268,258	0.01	8.09
150	721,030	0.66	54.44	45.56	8	6	10,805	14,946,664	0.01	23.66
200	966,000	3.35	62.75	37.25	13	9	10,800	31,751,134	0.01	37.31
Average		1.22	57.73	42.27			10,802		0.00	18
|K|	Exact Method on the minimizing/maximizing emissions problem									
	Minimizing Z_e							Maximizing Z_e		
	UB	Gap(%)	Z^{oe}(%)	Z^{te}(%)	p	q	CPU(s)	UB	Gap(%)	CPU(s)
50	3,891	0.57	61.87	38.13	17	14	10,837	50,388	0.00	0.95
100	9,611	1.68	51.50	48.50	18	14	10,816	202,808	0.01	2.75
150	13,320	1.86	55.74	44.26	18	14	10,825	424,761	0.00	5.24
200	26,089	3.34	75.89	24.11	18	14	10,807	1,769,814	0.00	32.21
Average		1.86	61.25	38.75			10,821		0.00	10

on the number of open DCs and RCs required, which is expressed as follows:

$$\bar{\mu} = \varepsilon \cdot \left\lfloor \frac{\sum_{k\in K}}{\max_{h\in H} \bar{b}_h} \right\rfloor \text{ and} \ddot{\mu} = \varepsilon \cdot \left\lfloor \frac{\sum_{k\in K}}{\max_{r\in R} \bar{b}_r} \right\rfloor, \tag{6.3}$$

where ε is a parameter set to 1.5. Those parameters are chosen based on preliminary experiments.

Tables 3 shows the summary of computational results in obtaining compromise solutions using the exact method (EM) and the matheuristic. Here, the proposed matheuristic is divided into two types, namely matheuristic with the interchange heuristic (MTH-ICH) and with the VNS (MTH-VNS). The first column of Table 3 refers to the number of customers. The table is mainly divided into two parts which are the results of minimizing L_1 and L_∞ problems. For each problem, the table also presents the solution obtained by EM, MTH-ICH and MTH-VNS represented by the deviation (%) achieved by the corresponding method together with its computational time (CPU). The best objective function value (Z^b) is also provided. The bold numbers in the table refer to the best solutions found.

According to Table 3, within 3 h, CPLEX was not able to guarantee optimality for the minimising L_1 and L_∞ problems. It is also noted that compared with the proposed matheuristic, CPLEX produced better solution for one instance only (i.e. $|K| = 50$ for the minimizing L_1 problem). It is worthwhile noting that the EM experienced difficulties when solving the minimizing L_1 and L_∞ problems, especially when $|K| > 50$. Based on the average deviation, the proposed matheuristic performs much better than the EM in obtaining the compromise solutions. The MTH-VNS provides a relatively small average deviation of 1.67 and 4.56% for the minimizing L_1 and L_∞ problems, respectively, whereas the EM yields approximately a massive value of 967 and 6,605%. Note that the large values of Dev (%) in Table 3 are mostly very large for the EM. This is due to the fact that CPLEX could not improve the UB within the maximum computing time of 3 h. Interesting results were observed where the MTH-ICH performs better for small instances (i.e. $|K| = 50$ and 100), whereas the MTH-VNS produces better results for the large ones (i.e. $|K| = 150$ and 200). It is mainly because MTH-VNS explores more feasible solutions rather than MTH-ICH. In summary, the matheuristic, especially MTH-VNS, is found to be the best method for generating good compromise solutions while consuming a smaller amount of computational effort.

Table 3 also presents the details of best compromise solutions obtained by the proposed methods, where the breakdown of the total cost and amount of emissions obtained are provided. Moreover, the information on the number of open DCs and RCs is given. It is worthwhile noting that the compromise solutions attained by the proposed are quite close to each other. For example, compromise solutions are shown in Fig. 2 for $|K| = 50$ and $|K| = 150$. The figure also reveals the ideal and non-ideal solutions. Here, compromise solutions are bounded by solutions generated by solving L_1 and L_∞ problems. The decision-maker, based on his/her individual preferences, will choose one or a few solutions from this solution set.

6.2 *Sensitivity and robustness analysis*

Sensitivity analysis

A detailed analysis is performed to assess the effect of the weight factor on the change in the number of DCs and RCs together with the balance between environmental and economical considerations. As a platform for discussion, the experiment is performed with an instance with $|K| = 50$, with $\tilde{w} = 0.1, 0.2, ..., 0.9$ leading to a set of solutions as shown in Fig. 3. In this case, the solution found for each (minimizing L_1 or L_∞ problem with a different $\tilde{w}$) using the proposed matheuristic is considered. Here,

Table 3 *Summary of computational results in obtaining compromise solutions for the problems with equal weight (0.5)*

$\vert K\vert$	Minimizing L_1 problem						Minimising L_∞ problem					
	Z^b	EM (3 h)	MTH-ICH		MTH-VNS		Z^b	EM (3 h)	MTH-ICH		MTH-VNS	
		Dev(%)	Dev(%)	CPU(s)	Dev(%)	CPU(s)		Dev(%)	Dev(%)	CPU(s)	Dev(%)	CPU(s)
50	0.0186	**0.00**	0.80	334	5.09	220	0.0101	6.70	**0.00**	382	11.19	223
100	0.0110	12.85	**0.00**	777	1.60	562	0.0063	17.15	**0.00**	787	7.05	468
150	0.0088	54.74	3.96	1,162	**0.00**	898	0.0044	635.91	37.35	937	**0.00**	1,015
200	0.0041	3,803.33	5.20	1,529	**0.00**	2,227	0.0019	25,759.68	32.21	1,419	**0.00**	2,225
Average		967.73	2.49	950.30	1.67	976.90		6,604.86	17.39	881.32	4.56	982.69

$\vert K\vert$	Best Configuration for the L_1 problem								Best Configuration for the L_∞ problem							
	Z_c	Z^{oc}(%)	Z^{tc} (%)	Z_e	Z^{oe} (%)	Z^{te} (%)	p	q	Z_c	Z^{oc} (%)	Z^{tc} (%)	Z_e	Z^{oe} (%)	Z^{te} (%)	p	q
50	201,878	65.75	34.25	4,995	60.81	39.19	15	8	212,664	66.21	33.79	4,835	63.76	36.24	11	9
100	614,565	63.48	36.52	12,077	48.69	51.31	13	11	640,603	60.66	39.34	12,036	48.11	51.89	14	10
150	838,551	62.78	37.22	17,182	50.28	49.72	14	11	845,140	61.42	38.58	16,909	51.90	48.10	13	11
200	1,107,137	66.51	33.49	32,495	70.16	29.84	16	13	1,080,148	63.95	36.05	32,659	67.61	32.39	14	13
Average		64.63	35.37		57.49	42.51				63.06	36.94		57.84	42.16		

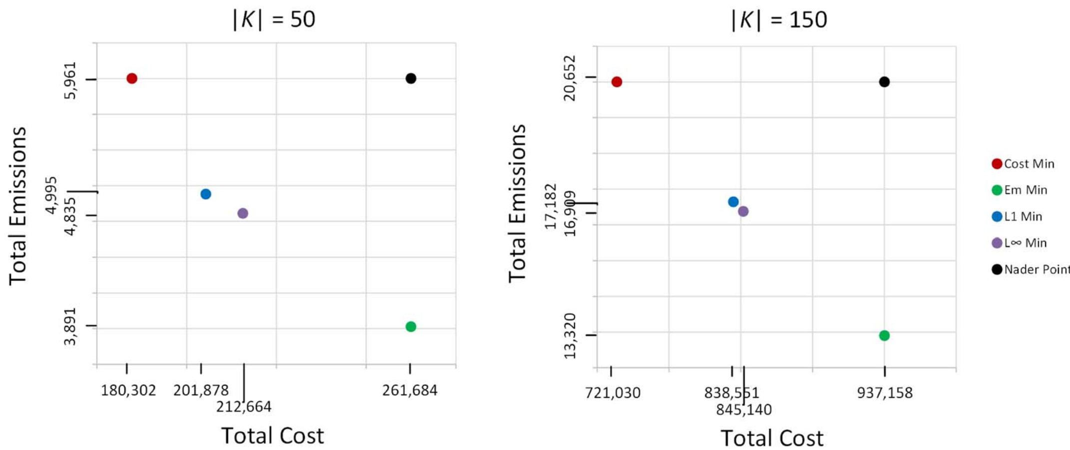


FIG. 2. Ideal, compromise and non-ideal solutions for $|K| = 50$ and $|K| = 150$.

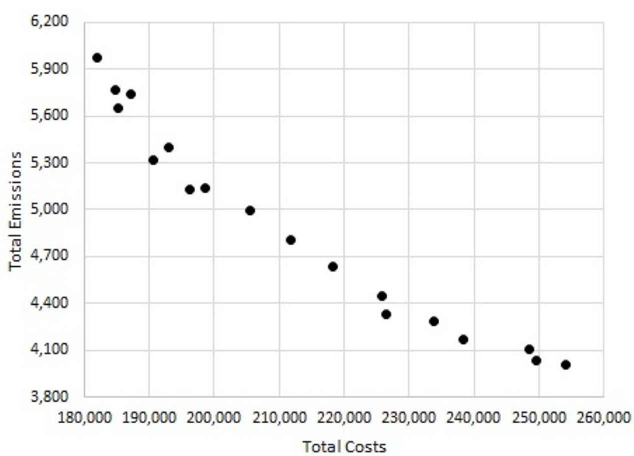
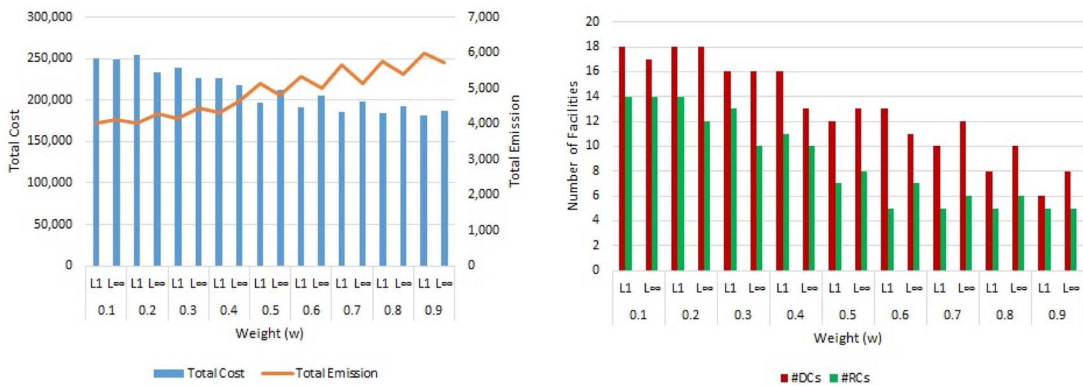


FIG. 3. A set of solutions for $|K| = 50$.

these generated solutions make up a Pareto frontier. When $\tilde{w}$ is set to a high value (e.g. $\tilde{w} = 0.9$), the total cost decreases but the total emissions increases. It is worth noting that, while the matheuristic is used to solve the minimizing L_1 or L_∞ problems, it is not guaranteed that the solutions produced are always non-dominated. This risk is linked to the heuristic nature and would obviously not happen if it was appropriate to apply an EM instead.

Figure 4 shows the breakdown solutions for each problem where Fig. 4a provides the total cost and emissions produced, whereas Fig. 4b presents the number of DCs and RCs. It is also highlighted that the required number of DCs and RCs decreases when the value of $\tilde{w}$ increases. This indicates that, despite the increased product movement (transportation) and consequent larger environmental impact, it is still economically efficient to have fewer DCs and RCs. For example, in the solution generated by solving



a) Total costs and emissions for each solution

b) The number of DCs and RCs for each solution

FIG. 4. The breakdown solutions for $|K| = 50$.

the L_1 problem with $\tilde{w} = 0.9$, there are 8 DCs and 5 RCs, which are selected. In this solution, the vehicle that provides the lowest cost for each open facility is chosen. On the other hand, if $\tilde{w}$ is small, say 0.1, we need to open 18 DCs and 14 RCs, with each facility requiring the vehicle, which produces the smallest amount of emissions. Solving the L_∞ problem with mid range values of $\tilde{w}$, such as $\tilde{w} = 0.4$ and 0.5, generates solutions that are located in the middle of the Pareto frontier. As an example, if $\tilde{w} = 0.5$ is used, 13 DCs and 8 RCs are required, whereas 13 DCs and 10 RCs are needed for $\tilde{w} = 0.4$. In this case, the type of vehicles used by each facility may be different. Even though this information is invaluable to senior management, the different configurations obtained presents a dilemma. We believe that the information given ought to be complemented by external factors, e.g. socio-economic information, in order for the management team to reach a compromise. Such an outcome could be based on a robust solution in order to remain financially and socially attractive for many years to come.

Robustness analysis

Here, we incorporate the concept of robustness to alleviate potential risks that could arise due to changes in some of the parameters, mentioned in the earlier subsection. It is worth emphasizing that decisions relating to the location and the capacity of the DCs and RCs, together with the transportation mode used by each facility, are strategic in nature. One way to address this complex decision issue is to provide a robust configuration for such problem. As an illustration, we investigate the robustness of a given configuration for a given scenario. The aim is to obtain a configuration that remains economically viable and environmentally attractive despite alterations in the input. Here, the presence of uncertain customer demand is analyzed using the SC configuration generated in the previous subsection. Monte Carlo simulation is designed and it is assumed that customer demand follows a normal distribution. The standard deviation of customer demand (σ_k) is determined based on the expected demand ($\tilde{d}_k$) with $\sigma_k = \psi \cdot \tilde{d}_k$, where ψ is a correction parameter and $\tilde{d}_k = d_k$.

The main procedure of the proposed simulation is presented in Algorithm 6.1 where the number of iterations ($\hat{T}$) needs to be defined first. For each iteration, the demand of customer ($\tilde{d}_k$) is randomly generated. The closed-loop supply chain assignment problem (CLSCAP) is solved using the EM within (τ''') seconds. The CLSCAP considers the uncertain demand ($\tilde{d}_k$) generated from the previous step. The decision variables used in the CLSCAP are the same as the ones given in Section 3, except that the decision variables $\bar{Y}_{jh}$, $\ddot{Y}_{lr}$, $\hat{U}^1_{iv}$, $\hat{U}^2_{jv}$, $\hat{U}^3_{lv}$ and $\hat{U}^4_{lv}$ are fixed. As the DCs and RCs have capacity constraints, new decision variables are introduced, representing the unmet demand of each customer for both the forward SC (δ_k^f) and the reverse SC (δ_k^b). In this study, the CLSCAP aims to minimize the overall unmet demand which is expressed as follows:

$$\min \sum_{k \in K} \left(\delta_k^f + \delta_k^b \right). \tag{6.4}$$

The constraints used by the CLSCAP are also similar to Equations (3.1)–(3.44) with minor modifications, where Constraints (3.14), (3.15), (3.20) and (3.26) are replaced by (6.5), (6.6), (6.7) and (6.8), respectively.

$$\sum_{k \in K} \sum_{v \in V^2} \left(U_{jkv}^2 \cdot (\tilde{d}_k - \delta_k^f) \right) \leq \sum_{h \in H} \left(\bar{Y}_{jh} \cdot \bar{b}_h \right), \qquad \forall j \in J \tag{6.5}$$

$$\sum_{i \in I} X_{ij}^1 = \sum_{k \in K} \sum_{v \in V^2} \left(U_{jkv}^2 \cdot (\tilde{d}_k - \delta_k^f) \right), \qquad \forall j \in J \tag{6.6}$$

$$\sum_{k \in K} \sum_{v \in V^3} \left(U_{klv}^3 \cdot (\tilde{d}_k - \delta_k^b) \cdot \alpha \right) \leq \sum_{r \in R} \left(\ddot{Y}_{lr} \cdot \ddot{b}_r \right), \qquad \forall l \in L \tag{6.7}$$

$$\sum_{k \in K} \sum_{v \in V^3} \left(U_{klv}^3 \cdot (\tilde{d}_k - \delta_k^b) \cdot \alpha \cdot \beta \right) \leq \sum_{i \in I} X_{li}^4, \qquad \forall l \in L. \tag{6.8}$$

As the CLSCAP is a non-linear model, we linearize it in the standard way so the model can be solved by a commercial solver such as CPLEX. Once $\hat{T}$ CLSCAP problems have been solved, the expected unmet demand for the forward SC ($\bar{\delta}_k^f$) and for the reverse SC ($\bar{\delta}_k^b$) are determined.

Algorithm 6.1 The simulation procedure

Define the number of iterations ($\hat{T}$)
for $\hat{t} = 1$ to $\hat{T}$ **do**
  Generate randomly the demand of customer ($\tilde{d}_k$) based on its distribution.
  Solve the CLSCAP using the exact method (CPLEX) within (τ''') seconds using ($\tilde{d}_k$). Store the decision variables obtained in the $\hat{t}^{\text{th}}$ iteration, namely the unmet demand of each customer for forward supply chain (δ_k^f) and for reverse supply chain (δ_k^b).
end for
Determine the expected total unmet demand for forward supply chain ($\bar{\delta}^f$) and for reverse supply chain ($\bar{\delta}^b$).

The experiments are conducted on an instance with $|K| = 50$, where two extreme solutions, as well as, two other solutions from the middle of the Pareto frontier given in Fig. 3 are selected. We vary the value of ψ from 0.1 to 0.3 with an increment of 0.1 in order to analyze if the standard deviation of customer demand influences the SC configuration. We set the value of $\hat{T}$ to 1,000 that represent

Table 4 *Summary of the computational results for the simulation*

Problem	$\tilde{w}$ (%)	ψ (%)	Out of 1,000 problem, #problems that have unmet demands	$\bar{\delta}_k^f$	$\bar{\delta}_k^b$
L_1	10	10	6	0.011	0
L_∞	40	10	34	0.09	0
L_∞	50	10	34	0.09	0
L_1	90	10	34	0.09	0
L_1	10	20	99	0.531	0.219
L_∞	40	20	175	1.091	0.074
L_∞	50	20	175	1.091	0.057
L_1	90	20	174	1.09	0.057
L_1	10	30	187	1.713	1.07
L_∞	40	30	255	2.632	0.621
L_∞	50	30	255	2.632	0.538
L_1	90	30	255	2.632	0.538

1,000 problems with different customer demands using an SC configuration. Here, we also analyze the number of problems that do not meet customer demands. Table 4 presents the summary of the computational results for this simulation. The SC configuration generated by solving an CLSC problem with a smaller $\tilde{w}$ (focusing on CO_2 emissions) resulted in a smaller number of cases which do not satisfy all of the customer demands. Also, as expected, the increase in the standard deviation (σ_k) will increase the number of cases that have unmet demands. We can state that our configuration is rather stable as long as the changes in customer demand are within a reasonable margin, say 10% of $\tilde{d}_k$. We believe that this finding is valuable to decision-makers as the analysis could be replicated onto their organizational scenarios, to eventually obtain an overall robust decision.

7. Conclusions and suggestions

In this study, we addressed a closed-loop SC and examined the challenging problem of sustainability using CP. We designed an optimisation model that incorporates two conflicting objectives, namely, the minimization of the total cost and the amount of CO_2 emissions. We modelled this closed-loop SC problem as a bi-objective mixed integer linear programming. The problem was solved to obtain the optimal number and locations of DCs and RCs, along with their capacity and the type of vehicle used. An effective matheuristic method, which is based on an aggregation technique, a reduced EM, an interchange-based heuristic and VNS, was then designed to overcome the difficulties faced by the original EM. The matheuristic technique was assessed using a variety of newly generated datasets which produced compromise solutions with higher quality than the ones found by the EM, while requiring only a fraction of the computing time.

To test the efficacy of our proposed methods, we performed scenario analysis followed by robustness analysis of the network configuration due to the changes in customer demand. The scenario analysis was to assess the effect of weight with respect to the objective functions which yielded different SC configurations. The robustness of the SC configuration is then assessed by applying Monte Carlo simulation on the customer demand. It was found that if the standard deviation is within 10% of the average demand, the unsatisfied demand is insignificant, thus demonstrating the stability of SC configuration. This invaluable information is key to assisting senior management to focus on attributes with impact on sustainability and resilience of their SC.

The following research directions may be worthy of future investigation. The uncertain demand of customers can be considered in the proposed model. Using a stochastic model instead of a deterministic one may be more difficult on a practical level, but it is more academically challenging to solve. In this study, only one product is considered; this restriction can be expanded to include a class of products instead. From a general viewpoint of heuristic search, other powerful metaheuristics including adaptive search methods could also be worth exploring.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

Abdulrahman, M. D., Gunasekaran, A. & Subramanian, N. (2014) Critical barriers in implementing reverse logistics in the chinese manufacturing sectors. *Int. J. Prod. Econ.*, **147**, 460–471.

Altmann, M. & Bogaschewsky, R. (2014) An environmentally conscious robust closed-loop supply chain design. *J. Bus. Econ.*, **84**, 613–637.

Asgari, N., Nikbakhsh, E., Hill, A. & Farahani, R. Z. (2016) Supply chain management 1982–2015: a review. *IMA J. Manag. Math.*, **27**, 353–379.

Asl-Najafi, J., Zahiri, B., Bozorgi-Amiri, A. & Taheri-Moghaddam, A. (2015) A dynamic closed-loop location-inventory problem under disruption risk. *Comput. Ind. Eng.*, **90**, 414–428.

Atabaki, M. S., Khamseh, A. A. & Mohammadi, M. (2019) A priority-based firefly algorithm for network design of a closed-loop supply chain with price-sensitive demand. *Comput. Ind. Eng.*, **135**, 814–837.

Atabaki, M. S., Mohammadi, M. & Naderi, B. (2020) New robust optimization models for closed-loop supply chain of durable products: Towards a circular economy. *Comput. Ind. Eng.*, **146**, 106520.

Chaabane, A., Ramudhin, A. & Paquet, M. (2012) Design of sustainable supply chains under the emission trading scheme. *Int. J. Prod. Econ.*, **135**, 37–49.

Chen, Y., Chan, F. & Chung, S. (2015) An integrated closed-loop supply chain model with location allocation problem and product recycling decisions. *Int. J. Prod. Res.*, **53**, 3120–3140.

Choudhary, A., Sarkar, S., Settur, S. & Tiwari, M. (2015) A carbon market sensitive optimization model for integrated forward-reverse logistics. *Int. J. Prod. Econ.*, **164**, 433–444.

CLHO. (2020) *Centre for Logistics and Heuristic Optimisation*. Canterbury, UK: Kent Business School, University of Kent. http://www.kent.ac.uk/kbs/research/research-centres/clho/datasets.html.

Correia, I., Nickel, S. & da Gama, F. S. (2010) Single-assignment hub location problems with multiple capacity levels. *Transp. Res. B Methodol.*, **44**, 1047–1066.

Dai, Z. & Zheng, X. (2015) Design of close-loop supply chain network under uncertainty using hybrid genetic algorithm: A fuzzy and chance-constrained programming model. *Comput. Ind. Eng.*, **88**, 444–457.

Dekker, R., Bloemhof, J. & Mallidis, I. (2012) Operations research for green logistics - an overview of aspects, issues, contributions and challenges. *Eur. J. Oper. Res.*, **219**, 671–679.

Demirel, N. O. & Gökçen, H. (2008) A mixed integer programming model for remanufacturing in reverse logistics environment. *Int. J. Adv. Manuf. Technol.*, **39**, 1197–1206.

Devika, K., Jafarian, A. & Nourbakhsh, V. (2014) Designing a sustainable closed-loop supply chain network based on triple bottom line approach: A comparison of metaheuristics hybridization techniques. *Eur. J. Oper. Res.*, **235**, 594–615.

Diabat, A. & Al-Salem, M. (2015) An integrated supply chain problem with environmental considerations. *Int. J. Prod. Econ.*, **164**, 330–338.

Elhedhli, S. & Merrick, R. (2012) Green supply chain network design to reduce carbon emissions. *Transp. Res. Part D: Transp. Environ.*, **17**, 370–379.

Fahimnia, B., Sarkis, J., Dehghanian, F., Banihashemi, N. & Rahman, S. (2013) The impact of carbon pricing on a closed-loop supply chain: an australian case study. *J. Clean. Prod.*, **59**, 210–225.

Fathollahi-Fard, A. M., Hajiaghaei-Keshteli, M. & Mirjalili, S. (2018) Hybrid optimizers to solve a tri-level programming model for a tire closed-loop supply chain network design problem. *Appl. Soft Comput.*, **70**, 701–722.

Fisher, M. L., Jaikumar, R. & Van Wassenhove, L. N. (1986) A multiplier adjustment method for the generalized assignment problem. *Manag. Sci.*, **32**, 1095–1103.

Fleischmann, M., Beullens, P., Bloemhof-Ruwaard, J. M. & Wassenhove, L. N. V. (2001) The impact of product recovery on logistics network design. *Prod. Oper. Manag.*, **10**, 156–173.

Ghahremani-Nahr, J., Kian, R. & Sabet, E. (2019) A robust fuzzy mathematical programming model for the closed-loop supply chain network design and a whale optimization solution algorithm. *Expert Syst. Appl.*, **116**, 454–471.

Ghisellini, P., Cialani, C. & Ulgiati, S. (2016) A review on circular economy: the expected transition to a balanced interplay of environmental and economic systems. *J. Clean. Prod.*, **114**, 11–32.

Gholizadeh, H., Jahani, H., Abareshi, A. & Goh, M. (2021) Sustainable closed-loop supply chain for dairy industry with robust and heuristic optimization. *Comput. Ind. Eng.*, **157**, 107324.

Govindan, K. & Soleimani, H. (2017) A review of reverse logistics and closed-loop supply chains: a journal of cleaner production focus. *J. Clean. Prod.*, **142**, 371–384.

Govindan, K., Soleimani, H. & Kannan, D. (2015) Reverse logistics and closed-loop supply chain: a comprehensive review to explore the future. *Eur. J. Oper. Res.*, **240**, 603–626.

Govindan, K., Darbari, J. D., Agarwal, V. & Jha, P. (2017) Fuzzy multi-objective approach for optimal selection of suppliers and transportation decisions in an eco-efficient closed loop supply chain network. *J. Clean. Prod.*, **165**, 1598–1619.

Hansen, P. & Mladenović, N. (1997) Variable neighborhood search for the p-median. *Locat. Sci.*, **5**, 207–226.

Huber, S., Cordeau, J. & Geiger, M. (2020) A matheuristic for the swap body vehicle routing problem. *OR Spectr.*, **42**, 111–160.

Irawan, C. A. & Jones, D. (2019) Formulation and solution of a two-stage capacitated facility location problem with multilevel capacities. *Ann. Oper. Res.*, **272**, 41–67.

Irawan, C. A., Salhi, S. & Drezner, Z. (2016) Hybrid meta-heuristics with vns and exact methods: application to large unconditional and conditional vertex p-center problems. *J. Heuristics*, **22**, 507–537. https://doi.org/10.1007/s10732-014-9277-7.

Irawan, C. A., Jones, D. & Ouelhadj, D. (2017a) Bi-objective optimisation model for installation scheduling in offshore wind farms. *Comput. Oper. Res.*, **78**, 393–407.

Irawan, C. A., Salhi, S., Luis, M. & Azizi, N. (2017b) The continuous single source location problem with capacity and zone-dependent fixed cost: Models and solution approaches. *Eur. J. Oper. Res.*, **263**, 94–107.

Irawan, C. A., Song, X., Jones, D. & Akbari, N. (2017c) Layout optimisation for an installation port of an offshore wind farm. *Eur. J. Oper. Res.*, **259**, 67–83.

Irawan, C., Wall, G. & Jones, D. (2019) An optimisation model for scheduling the decommissioning of an offshore wind farm. *OR Spectr.*, **41**, 513–548.

Jones, D. (2011) A practical weight sensitivity algorithm for goal and multiple objective programming. *Eur. J. Oper. Res.*, **213**, 238–245.

Kadambala, D. K., Subramanian, N., Tiwari, M. K., Abdulrahman, M. & Liu, C. (2017) Closed loop supply chain networks: Designs for energy and time value efficiency. *Int. J. Prod. Econ.*, **183**, 382–393.

Kalaitzidou, M. A., Longinidis, P. & Georgiadis, M. C. (2015) Optimal design of closed-loop supply chain networks with multifunctional nodes. *Comput. Chem. Eng.*, **80**, 73–91.

Kang, K., Wang, X. & Ma, Y. (2017) A collection-distribution center location and allocation optimization model in closed-loop supply chain for chinese beer industry. *Math. Probl. Eng.*, **2017**, 1. https://doi.org/10.1155/2017/7863202.

Kaveh, F., Tavakkoli-Moghaddam, R., Triki, C., Rahimi, Y. & Jamili, A. (2019) A new bi-objective model of the urban public transportation hub network design under uncertainty. *Ann. Oper. Res.*, **296**, 131–162. https://doi.org/10.1007/s10479-019-03430-9.

Kaya, O. & Urek, B. (2016) A mixed integer nonlinear programming model and heuristic solutions for location, inventory and pricing decisions in a closed loop supply chain. *Comput. Oper. Res.*, **65**, 93–103.

Kazancoglu, Y., Kazancoglu, I. & Sagnak, M. (2018) A new holistic conceptual framework for green supply chain management performance assessment based on circular economy. *J. Clean. Prod.*, **195**, 1282–1299.

Ko, H. J. & Evans, G. W. (2007) A genetic algorithm-based heuristic for the dynamic integrated forward/reverse logistics network for 3pls. *Comput. Oper. Res.*, **34**, 346–366.

Kusakci, A. O., Ayvaz, B., Cin, E. & Aydin, N. (2019) Optimization of reverse logistics network of end of life vehicles under fuzzy supply: A case study for istanbul metropolitan area. *J. Clean. Prod.*, **215**, 1036–1051.

Kusumastuti, R. D., Piplani, R. & Lim, G. H. (2008) Redesigning closed-loop service network at a computer manufacturer: A case study. *Int. J. Prod. Econ.*, **111**, 244–260.

Lee, D. H. & Dong, M. (2008) A heuristic approach to logistics network design for end-of-lease computer products recovery. *Transp. Res. E: Logist. Transp. Rev.*, **44**, 455–474.

Li, Y., Guo, H. & Zhang, Y. (2018) An integrated location-inventory problem in a closed-loop supply chain with third-party logistics. *Int. J. Prod. Res.*, **56**, 3462–3481.

Lu, Z. & Bostel, N. (2007) A facility location model for logistics systems including reverse flows: The case of remanufacturing activities. *Comput. Oper. Res.*, **34**, 299–323.

Mota, B., Gomes, M. I., Carvalho, A. & Barbosa-Povoa, A. P. (2015) Towards supply chain sustainability: economic, environmental and social design and planning. *J. Clean. Prod.*, **105**, 14–27.

Nayeri, S., Paydar, M. M., Asadi-Gangraj, E. & Emami, S. (2020) Multi-objective fuzzy robust optimization approach to sustainable closed-loop supply chain network design. *Comput. Ind. Eng.*, **148**, 106716.

Obal, T. M., Irawan, C. A., Jones, D., Ouelhadj, D., Florentino, H. O., Gryczak, V., Volpi, N. M. P. & Wilhelm, V. E. (2019) Development and evaluation of a matheuristic for the combined beam angle and dose distribution problem in radiotherapy planning. *IMA J. Manag. Math.*, **30**, 413–430.

OECD/IEA. (2017) CO_2 emissions from fuel combustion 2017 - highlights. https://webstore.iea.org/co2-emissions-from-fuel-combustion-highlights-2017 (accessed 6 December 2017).

Oliveira, L. S. & Machado, R. L. (2021) Application of optimization methods in the closed-loop supply chain: a literature review. *J. Comb. Optim.*, **41**, 357–400.

Pan, S., Ballot, E. & Fontane, F. (2013) The reduction of greenhouse gas emissions from freight transport by pooling supply chains. *Int. J. Prod. Econ.*, **143**, 86–94.

Papen, P. & Amin, S. H. (2019) Network configuration of a bottled water closed-loop supply chain with green supplier selection. *J. Remanufacturing*, **9**, 109–127.

Patne, K., Shukla, N., Kiridena, S. & Tiwari, M. K. (2018) Solving closed-loop supply chain problems using game theoretic particle swarm optimisation. *Int. J. Prod. Res.*, **56**, 5836–5853.

Pishvaee, M. S. & Razmi, J. (2012) Environmental supply chain network design using multi-objective fuzzy mathematical programming. *Appl. Math. Model.*, **36**, 3433–3446.

Pishvaee, M. & Torabi, S. (2010) A possibilistic programming approach for closed-loop supply chain network design under uncertainty. *Fuzzy Sets Syst.*, **161**, 2668–2683.

Pishvaee, M. S., Farahani, R. Z. & Dullaert, W. (2010) A memetic algorithm for bi-objective integrated forward/reverse logistics network design. *Comput. Oper. Res.*, **37**, 1100–1112.

Ramezani, M., Kimiagari, A. M., Karimi, B. & Hejazi, T. H. (2014) Closed-loop supply chain network design under a fuzzy environment. *Knowl.-Based Syst.*, **59**, 108–120.

Ramos, T., Gomes, M. & Barbosa-Póvoa, A. (2020) A new matheuristic approach for the multi-depot vehicle routing problem with inter-depot routes. *OR Spectr.*, **42**, 75–110.

Romero, C. & Rehman, T. (1989) *Multiple Criteria Analysis for Agricultural Decisions*. Amsterdam, The Netherland: Elsevier Science Publishers.

Salehi-Amiri, A., Zahedi, A., Akbapour, N. & Hajiaghaei-Keshteli, M. (2021) Designing a sustainable closed-loop supply chain network for walnut industry. *Renew. Sust. Energ. Rev.*, **141**, 110821.

Salema, M. I. G., Póvoa, A. P. B. & Novais, A. Q. (2009) A strategic and tactical model for closed-loop supply chains. *OR Spectr.*, **31**, 573–599.

Salhi, S., 2017. *Heuristic Search: The Emerging Science of Problem Solving*. Springer, Cham, Switzerland. https://doi.org/10.1007/978-3-319-49355-8.

Salhi, S. & Rand, G. K. (1989) The effect of ignoring routes when locating depots. *Eur. J. Oper. Res.*, **39**, 150–156.

Song, X., Jones, D., Asgari, N. & Pigden, T. (2020) Multi-objective vehicle routing and loading with time window constraints: a real-life application. *Ann. Oper. Res.*, **291**, 799–825.

Sridharan, R. (1995) The capacitated plant location problem. *Eur. J. Oper. Res.*, **87**, 203–213.

Subramanian, P., Ramkumar, N., Narendran, T. & Ganesh, K. (2013) Prism: Priority based simulated annealing for a closed loop supply chain network design problem. *Appl. Soft Comput.*, **13**, 1121–1135.

Talaei, M., Moghaddam, B. F., Pishvaee, M. S., Bozorgi-Amiri, A. & Gholamnejad, S. (2016) A robust fuzzy optimization model for carbon-efficient closed-loop supply chain network design problem: a numerical illustration in electronics industry. *J. Clean. Prod.*, **113**, 662–673.

Tiwari, A., Chang, P. C., Tiwari, M. & Kandhway, R. (2016) A hybrid territory defined evolutionary algorithm approach for closed loop green supply chain network design. *Comput. Ind. Eng.*, **99**, 432–447.

Tosarkani, B. M. & Amin, S. H. (2018) A possibilistic solution to configure a battery closed-loop supply chain: Multi-objective approach. *Expert Syst. Appl.*, **92**, 12–26.

Wang, H. F. & Hsu, H. W. (2010) A closed-loop logistic model with a spanning-tree based genetic algorithm. *Comput. Oper. Res.*, **37**, 376–389.

Wang, K., Shao, Y. & Zhou, W. (2017) Matheuristic for a two-echelon capacitated vehicle routing problem with environmental considerations in city logistics service. *Transp. Res. Part D: Transp. Environ.*, **57**, 262–276.

Yadegari, E., Alem-Tabriz, A. & Zandieh, M. (2019) A memetic algorithm with a novel neighborhood search and modified solution representation for closed-loop supply chain network design. *Comput. Ind. Eng.*, **128**, 418–436.

Yakavenka, V., Mallidis, I., Vlachos, D., Iakovou, E. & Eleni, Z. (2019) Development of a multi-objective model for the design of sustainable supply chains: the case of perishable food products. *Ann. Oper. Res.*, **294**, 593–621.

Yi, P., Huang, M., Guo, L. & Shi, T. (2016) A retailer oriented closed-loop supply chain network design for end of life construction machinery remanufacturing. *J. Clean. Prod.*, **124**, 191–203.

Zhen, L., Sun, Q., Wang, K. & Zhang, X. (2019) Facility location and scale optimisation in closed-loop supply chain. *Int. J. Prod. Res.*, **57**, 7567–7585.