Babaveisi, Vahid; Paydar, Mohammad Mahdi; Safaei, Abdul Sattar

Article

Optimizing a multi-product closed-loop supply chain using NSGA-II, MOSA, and MOPSO meta-heuristic algorithms

Journal of Industrial Engineering International

Provided in Cooperation with:
Islamic Azad University (IAU), Tehran

Suggested Citation: Babaveisi, Vahid; Paydar, Mohammad Mahdi; Safaei, Abdul Sattar (2018) : Optimizing a multi-product closed-loop supply chain using NSGA-II, MOSA, and MOPSO meta-heuristic algorithms, Journal of Industrial Engineering International, ISSN 2251-712X, Springer, Heidelberg, Vol. 14, Iss. 2, pp. 305-326, http://dx.doi.org/10.1007/s40092-017-0217-7

This Version is available at:
http://hdl.handle.net/10419/195609

Terms of use:
Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.

https://creativecommons.org/licenses/by/4.0/
Optimizing a multi-product closed-loop supply chain using NSGA-II, MOSA, and MOPSO meta-heuristic algorithms

Vahid Babaveisi1 · Mohammad Mahdi Paydar1 · Abdul Sattar Safaei1

Received: 13 March 2017 / Accepted: 16 June 2017 / Published online: 1 July 2017
© The Author(s) 2017. This article is an open access publication

Abstract This study aims to discuss the solution methodology for a closed-loop supply chain (CLSC) network that includes the collection of used products as well as distribution of the new products. This supply chain is presented on behalf of the problems that can be solved by the proposed meta-heuristic algorithms. A mathematical model is designed for a CLSC that involves three objective functions of maximizing the profit, minimizing the total risk and shortages of products. Since three objective functions are considered, a multi-objective solution methodology can be advantageous. Therefore, several approaches have been studied and an NSGA-II algorithm is first utilized, and then the results are validated using an MOSA and MOPSO algorithms. Priority-based encoding, which is used in all the algorithms, is the core of the solution computations. To compare the performance of the meta-heuristics, random numerical instances are evaluated by four criteria involving mean ideal distance, spread of non-dominance solution, the number of Pareto solutions, and CPU time. In order to enhance the performance of the algorithms, Taguchi method is used for parameter tuning. Finally, sensitivity analyses are performed and the computational results are presented based on the sensitivity analyses in parameter tuning.

Keywords Closed-loop logistics · NSGA-II · MOPSO · MOSA · Taguchi · Mathematical model · Priority-based

Introduction

The incremental trend of industrialization demands more and more resources for producing the innovative and old-fashioned products. However, two main types of resources can be identified. Renewable resources are sources of energy that are not depleted by use, such as wind, or solar power and the like. A non-renewable resource cannot be replaced by the nature such as fossil fuels. In this study, we focus on the non-renewable resources that are noteworthy due to depleting. Considering the increase in demand for resources, these resources will be too expensive as they become less available. Consequently, many companies try to produce products that use resources supplied by the end-user. In this chain, the end-users deliver the used products to collection centers to be used for producing products in manufacturing centers (McWilliams and Siegel 2001). The process of collecting products from the end-users and recycling is called “reverse logistics” (Chen et al. 2017). A good instance for reverse logistics is the Japanese Ministry of the Environment for the automotive industry (Sudarto et al. 2016). Recent studies show a good classification for the investigated industries related to CLSC. These studies involve 69 papers between 2003 and 2015 (Govindan and Soleimani 2017). As shown in Table 1, no study is performed in the area of oil and related fields during 12 years. This reason encouraged us to make a research in this area.

Oil involves many derivatives such as kerosene, lubricant, gasoline, and tar. In this study, we concentrate on the lubricants. Engine oil that is one of the important products which is used in cars for lubrication of engine parts. Since oil is one of the expensive and non-renewable products, designing an optimized reverse logistics will be advantageous both financially and environmentally (Özceylan and Paksoy 2013).

This paper aims to design a closed-loop reverse logistics for collecting used engine oil that will be used in manufacturing
centers. Most of real-world supply chain (SC) problems are complex due to high number of indices which increase the dimension of the problem and it may lead to inefficiency of routine solution approaches (Fahimnia et al. 2013). Regarding increase in the size of problem, exponential growth in complexity makes the model become NP-hard (Park et al. 2007; Jolai et al. 2011). To deal with this problem, meta-heuristic solution methodologies are applied in this study. The algorithms that have been employed in this study are NSGA-II, MOSA, and MOPSO. In order to validate the results generated by each algorithm, three meta-heuristics are used.

The paper is composed of the following sections. In “Literature review,” a comprehensive study is performed to examine earlier researches and introduce the present research path. The problem is described and the mathematical model is represented in “Problem statement” section. Meta-heuristics are presented in the “Solution methodology”. The criteria are introduced in “Performance measure.” Taguchi approach is discussed in “Parameter setting,” and finally, brief results of the study and future research recommendations are provided in “Conclusion and future research” section.

### Literature review

In this section, solution methodologies for solving mathematical models are presented specifically supply chain network models which were proposed by recent researchers. Since the current work focused on the meta-heuristic solution approaches the field of the supply chain, the concerned papers are discussed as follows.

Bandyopadhyay et al. (2008) proposed a new simulated annealing (SA) called archived multi-objective simulated annealing (AMOSA). The proposed algorithm was compared with two other solution methods: (1) non-dominated sorting genetic algorithm (NSGA-II) (2) Pareto archived evolution strategy (PAES). The results showed better performance of AMOSA than NSGA-II and PAES. Mahdavi et al. (2009) designed a model for a cell formation problem in cellular manufacturing was designed for this problem. Genetic algorithm was used for solving the model. Simple and uniform crossovers as well as a heuristic mutation generate new solutions. A genetic algorithm was implemented for optimization that showed better results than other methods.

| Case study                   | No. | Articles                                                                 |
|------------------------------|-----|---------------------------------------------------------------------------|
| Without case study           | 14  | Govindan et al. (2013); Özkir and Başgil (2013); Nikolaou et al. (2013); Bai and Sarkis (2013); Rashid et al. (2013); Dwivedy and Mittal (2012); Pialot et al. (2012); Chen and Sheu (2013); Gerrard and Kandlikar (2007); Hatcher et al. (2011); Binnemans et al. (2013); Qiang (2015); Corum et al. (2014); Chen et al. (2014) |
| Car parts suppliers          | 11  | Subramoniam et al. (2009); Matsumoto (2010); Subramoniam et al. (2010); Hojas et al. (2011); Subramoniam et al. (2013); Blume and Walthier (2013); Amelia et al. (2009); Abdulrahman et al. (2014); Xia et al. (2014); Wang et al. (2014); Tian et al. (2014) |
| Vehicle manufacturer/remanufacturer | 8 | Seitz (2007); Saavedra et al. (2013); McKenna et al. (2013); Forslind (2005); Go et al. (2011); Demirel et al. (2014); Kurdie et al. (2015); Ziout et al. (2014) |
| Electronics                  | 8   | Georgiadis and Besiou (2008); Mafakheri and Nasiri (2013); Rathore et al. (2011); White et al. (2003); El korchi and Millet (2011); Ravi (2012); Low et al. (2014); Jiménez-Parra et al. (2014) |
| Steel                        | 3   | Giannetti et al. (2013); Schultmann et al. (2004); Salmi and Wierink (2011) |
| Tannery                      | 2   | Hu et al. (2011); Gutterres et al. (2010) |
| Power                        | 2   | Ortegon et al. (2013); Jiang et al. (2011) |
| Printing                     | 2   | Kerr and Ryan (2001); Cristóbal Andrade et al. (2012) |
| Palm oil                     | 2   | Ng et al. (2012); Mumtaz et al. (2010) |
| Machine tools                | 2   | Silva et al. (2013); Du et al. (2012) |
| Nutrition                    | 2   | Huysveld et al. (2013); Jefferies et al. (2012) |
| Others                       | 13  | Fahimnia et al. 2013; Simpson (2012); Zhu and Geng (2013); Matsumoto (2009); Zeng et al. (2013); Östlin et al. (2009); Pigosso et al. (2010); Queiruga et al. (2012); Shaharudin et al. (2014); Murakami et al. (2014); Goodall et al. (2014); Gelbmann and Hammerl (2014); Song et al. (2014) |
Table 2: A brief of reviewed meta-heuristic algorithm

| Author(s)                        | Implemented meta-heuristic algorithm | Field of the research                          |
|----------------------------------|--------------------------------------|-----------------------------------------------|
|                                  | Memetic GA NSGA-II NRGA ABC Hybrid GA SA MOPSO MOGA MOSA |                                                |
| Bandyopadhyay et al. (2008)      |                                      | Theoretical discussion                         |
| Pishvae et al. (2009)            |                                      | Reverse logistics                              |
| Lee et al. (2009)                |                                      | Reverse logistics                              |
| Mahdavi et al. (2009)            |                                      | Cellular manufacturing                         |
| Altiparmak et al. 2009           |                                      | Forward logistics                              |
| Pishvae et al. 2010              |                                      | Forward/reverse logistics                      |
| Umar et al. (2012)               |                                      | Vehicle routing                               |
| Lotfi and Tavakkoli-Moghaddam (2013) |                                      | Transportation problem                         |
| Yang et al. (2013)               |                                      | Line balancing model                           |
| Subramanian et al. (2013)        |                                      | Closed-loop supply chain                       |
| Ghasimi et al. (2014)            |                                      | Forward logistics                              |
| Yadegari et al. (2015)           |                                      | Forward/reverse logistics                      |
| Pasandideh et al. (2015a, b)     |                                      | Forward logistics                              |
| Santosa and Kresna (2015)        |                                      | Facility location problem                      |
| Liu et al. (2015)                |                                      | Oil–gas production                             |
| Sarraha et al. (2015)            |                                      | Forward logistics                              |
| Kumar et al. (2015)              |                                      | Reverse logistics                              |
| Mousavi et al. (2016)            |                                      | Inventory model                               |
| Current study                    |                                      | Closed-loop supply chain                       |
|                                  |                                      |                                                |

SC. The proposed model was solved using a hybrid genetic algorithm. A novel method is used for crossover to generate elite offsprings called weight mapping crossover (WMX). Pishvae et al. (2010) designed a bi-objective model for a SC which involves two objective functions of maximizing responsiveness and minimizing costs. Due to insufficient search capability of genetic algorithm, a memetic algorithm is used for local search improvement. Since two objective functions are defined in their model, a multi-objective algorithm is implemented for optimizing the proposed model. Umar et al. (2012) discussed a vehicle routing problem (VRP) which was formulated as a single-objective mathematical model. Since GA was implemented for solving this problem, the priority-based method was used for genetic representation. Subramanian et al. (2013) considered a CLSC network, and then the simulated annealing algorithm was conducted in the proposed CLSC using a priority-based encoding for chromosome representation. Yang et al. (2013) considered a mixed-model assembly line which contains a multi-objective function. To solve this model, an extended version of the genetic algorithm is used, called multi-objective genetic algorithm (MOGA). Partial representation is implemented for chromosome representation. Lotfi and Tavakkoli-Moghaddam (2013) discussed a transportation problem which was solved using a genetic algorithm. New operators of crossover and mutation were proposed in their paper. The priority-based algorithm was tested against spanning-tree-based algorithm by different instances of small, medium and large-sized problems that outperformed for medium and large-sized problems. Ghasimi et al. (2014) proposed a model for a defective goods SC network with an objective function of minimizing costs. This model determines the appropriate length of each cycle based on the importance of customer lead time using just-in-time logistics. Finally, the results of CPLEX solver and the GA are compared that CPLEX showed better results. The chromosomes are displayed in a matrix representing quantity variable. Yadegari
et al. (2015) implemented a memetic algorithm for an NP-hard problem of an integrated forward-reverse logistics network. Random path-based direct decoding procedure is used for the delivery and recovery path generation. Pasandideh et al. (2015) implemented two bi-objective optimization methods of NRGA and NSGA-II to solve the model for a SC network. Priority-based was used for chromosome representation. NSGA-II for outperformed NRGA in this problem. Santosa and Kresna (2015) designed a mathematical model for a warehouse location problem. Simulated annealing was used for solving the model. Quantity of variables construct the chromosome matrix. An 11% gap exists between exact and metaheuristic methods. Since computation time is more important, SA outperforms the exact method. Liu et al. (2015) designed a multi-objective model. Two maximizing and one minimizing objective functions are used. To solve the model, a modified version of NSGA-II called I-NSGA-II is implemented which enhances diversity and convergence of the solutions. Sarrafha et al. (2015) designed a SC that was formulated as an MILP model. A multi-objective biogeography-based optimization (MOBBO) was utilized as a solution methodology. Habitat structure was used to represent the chromosome. The results of their method are compared with NSGA-II and MOSA algorithms. Kumar et al. (2015) designed a mathematical model for a reverse logistics. To solve the model, an artificial bee colony approach was utilized. Mousavi et al. (2016) discussed a bi-objective mathematical model for an inventory optimization problem. The reviewed papers are classified in Table 2, according to the implemented algorithms.

Among reviewed papers, the papers that utilized priority-based representation are classified according to the type of the proposed supply chain and its specifications in Table 3.

Based on the reviewed papers, none of the studies have been performed on the closed-loop supply chains considering multi-objective, multi-product, and multi-period models. Current paper considers a CLSC for collecting used engine oil that is multi-product, multi-period, and multi-objective. In this study, the NSGA-II, MOSA, and MOPSO multi-objective solution methods are utilized to get the optimal solution for the proposed multi-objective mathematical model.

**Problem statement**

Used engine oil collection involves several layers that have been studied practically to obtain information for designing the supply chain network. The flow of collected used oil starts from vendors and continues to collection centers and then to the manufacturing centers. Two types of manufacturing centers are considered. Type one is manufacturing centers using the original oil to produce product type one; the second type is manufacturing centers that purify used engine oil and use it in the type two product. In the forward supply chain, products are distributed from

| Author(s)          | Objective function | Product(s)          | Period(s)          | Type of supply chain |
|--------------------|--------------------|---------------------|--------------------|----------------------|
| Pishvae et al.     | *                  | *                   |                    |                      |
| Lee et al. (2009)  | *                  | *                   |                    |                      |
| Altiparmak et al.  | *                  | *                   |                    |                      |
| Pishvae et al.     | *                  | *                   |                    |                      |
| Subramanian et al. | *                  | *                   |                    |                      |
| Yadegari et al.    | *                  | *                   |                    |                      |
| Pasandideh et al.  | *                  | *                   | *                  | *                    |
| Sarrafha et al.    | *                  | *                   | *                  | *                    |
| Current study      | *                  | *                   | *                  | *                    |

Table 3: The main classification of supply chain papers considering priority-based representation
manufacturing centers to distribution centers and finally to vendors. Described CLSC is presented in Fig. 1.

Assumptions, notations, and parameters

- Manufacturing centers are two types. The first one is manufacturing type one that produces products type one using original oil and the second one is manufacturing type two which produces products by used engine oil.
- Some hybrid centers are defined which function as distribution and collection centers.
- Used engine oil is supplied by vendors and original oil is provided by the suppliers.
- The capacity of containers that is used for transferring the used engine oil from vendors to collection centers is \( PO_1 \).
- Vehicles which transfer oil from collection centers to manufacturing centers have a capacity of \( PO_2 \).

Considering the defined problem and its assumptions, a model is designed for the proposed SC. The notations, parameters, and decision variables are as follows:

Indices

| Symbol | Description |
|--------|-------------|
| \( c \) | Index of the first type of products; \( c = 1, \ldots, C \) |
| \( d \) | Index of the second type of products; \( d = 1, \ldots, D \) |
| \( r \) | Index of products; \( r = c \cup d \) |
| \( i \) | Index of the first type of manufacturing plants (MPs); \( i = 1, \ldots, I \) |
| \( j \) | Index of the second type of MPs; \( j = 1, \ldots, J \) |
| \( p \) | Index of MPs; \( p = i \cup j \) |

Parameters

| Symbol | Description |
|--------|-------------|
| \( C_{\text{rpt}} \) | Production cost related to product \( r \) which is produced in MP \( p \) in period \( t \) |
| \( \text{WEI}_r \) | Weight of product \( r \) |
| \( TC_{1, pkt} \) | Cost of shipping product \( r \) from MP \( p \) to DC \( k \) in period \( t \) |
| \( TC_{2, kvt} \) | Cost of shipping product \( r \) from DC \( k \) to vendor \( v \) in period \( t \) |
| \( TC_{3, k'0} \) | Cost of shipping one unit of used engine oil from vendor \( v \) to CC \( k' \) in period \( t \) |
| \( TC_{4, k'0} \) | Cost of shipping one unit of used engine oil from CC \( k' \) to MP \( p \) in period \( t \) |
| \( \text{Cap}_{\text{rpt}} \) | Capacity of MP \( r \) for product \( p \) in period \( t \) |
| \( \text{Cap}_{\text{kt}} \) | Capacity of DC \( k \) in period \( t \) |
| \( \text{Cap}_{\text{000} k'0} \) | Number of available \( PO_2 \) containers in CC \( k' \) in period \( t \) |
| \( \text{Cap}_{\text{v0}} \) | Capacity of vendor \( v \) to supply used oil in period \( t \) |
| \( \text{BE}_{vt} \) | Purchase cost for each \( PO_1 \) container of used oil from vendor \( v \) in period \( t \) |
| \( PP_{rpt} \) | Price of product \( r \) in MP \( p \) in period \( t \) |
| \( B_{rt} \) | Shortage cost of product \( r \) in period \( t \) |
| \( \tau_r \) | Amount of oil to produce one unit of product \( r \) |
| \( \text{PO1} \) | Weight of small container to ship used oil from vendors to collection and hybrid centers |
| \( \text{PO2} \) | Weight of each tanker of used engine oil shipped from CC to MP |
| \( D_{\text{vrt}} \) | Demand of vendor \( v \) for product \( r \) in period \( t \) |
| \( R_{k'v} \) | Rate of risk of supplying used oil by vendor \( v \) to CC \( k' \) |
| \( EC_e \) | Establishment cost of DC of \( e \) |

Fig. 1 Illustration of the proposed closed-loop logistics
Max \( Z_1 = \text{INCOMES} - \text{COSTS} \)  

Decision variables

\( Q_{rpt} \) Quantity of production of product \( r \) in MP \( p \) in period \( t \)  
\( X_{rpt} \) Quantity of product \( r \) shipped from MP \( p \) to DC \( k \) in period \( t \)  
\( Y_{vkt} \) Quantity of product \( r \) shipped from DC \( k \) to vendor \( v \) in period \( t \)  
\( Y'_{vkt} \) Amount of used oil transferred from vendor \( v \) to CC \( k \) in period \( t \)  
\( X'_{kpt} \) Amount of used oil transferred from CC \( k' \) to MP \( p \) in period \( t \)  
\( L_{rvt} \) Shortage of product \( r \) for vendor \( v \) in period \( t \)  
\( DES_e \) If DC \( e \) is open 1, otherwise 0  
\( CES_f \) If CC \( f \) is open 1, otherwise 0  
\( HES_h \) If HC \( h \) is open 1, otherwise 0

Mathematical model

The first objective function (1) is maximizing total profit. It is obtained by SC costs from total incomes.

\[ \text{Max } Z_1 = \text{INCOMES} - \text{COSTS} \]  

Term (2) shows incomes obtained from selling products in the forward direction of the SC network.

\[ \text{INCOMES} = \sum_r \sum_p \sum_t \sum_k \sum_v PP_{rpt} Y_{vkt} \]  

Terms (3) and (4) show facility establishment and production costs, respectively.

\[ \text{COSTS} = \sum_e EC_e DES_e + \sum_f EC'_f CES_f + \sum_h EC''_h HES_h + \sum_r \sum_p \sum_t C_{rpt} Q_{rpt} \]  

Transportation costs between facilities are obtained from term (5).

\[ + \sum_r \sum_p \sum_k \sum_t TC1_{rpt} X_{rpt} \times WEI \times X_{rpt} \]  
\[ + \sum_r \sum_k \sum_t TC2_{rkt} X_{rkt} \times WEI \times Y_{rkt} \]  
\[ + \sum_r \sum_k \sum t TC3_{vkt} \times PO1 \times Y'_{vkt} \]  
\[ + \sum_k \sum p \sum t TC4_{kpt} \times PO1 \times X'_{kpt} \]  

Terms (6) and (7) represent used engine oil purchase cost and forward products shortage costs, respectively.

\[ + \sum_r \sum_k \sum t BE_{rvt} \times [Y'_{vkt}] \]  
\[ + \sum_r \sum_v \sum t B_{rvt} L_{rvt} \]  

Minimization of total risks of purchasing used oil from vendors is represented as the second objective function in Eq. (8).

\[ \text{Min } Z_2 = \sum_r \sum_k \sum_t RI_{rvt} \times Y'_{vkt} \]  

Finally, minimizing the shortage of products is obtained from the third objective function in term (9).

\[ \text{Min } Z_3 = \sum_r \sum t \left( \sum_v L_{rvt} \right) \]  

Constraints

The number of products transferred from manufacturing centers to distribution centers should equal the quantity of production in manufacturing centers (10).

\[ Q_{rpt} - \sum_k X_{rpt} = 0 \quad \forall r, p, t \]  

Constraints (11) present the balance of flow in distribution centers. To generate a feasible solution, it is mandatory that output from DCs be less than or equal to input to DCs.

\[ \sum_p X_{rpt} - \sum_v Y_{vkt} = 0 \quad \forall r, k, t \]  

Constraint (12) presents shortages of forward products. If the number of transferred products to vendors are less than their demands, there will be shortages of products.

\[ \sum_k Y_{vkt} + L_{rvt} = D_{rvt} \quad \forall r, v, t \]  

The balance of flow in collection centers is specified using constraint (13).

\[ PO1 \sum_v Y'_{vkt} - PO2 \left( \sum_r X'_{kpt} \right) = 0 \quad \forall k', t \]  

Maximum allowed production in manufacturing centers is bounded by their capacities in term (14). Capacity is defined for each product \( r \) in each manufacturing center \( p \) in period \( t \).

\[ Q_{rpt} \leq \text{cap}_{rpt} \quad \forall r, p, t \]  

Each vendor has a certain capacity to supply used engine oil that is shown in constraint (15). This capacity is the amount of used engine oil that can be provided by each vendor in each period as the resources of producing the second type of products.
PO1 \times \sum_{k'} Y_{v'k't}^{v'} \leq cap_{vt}^{v'} \quad \forall v, t \tag{15}

Collection centers have a capacity of processing used engine oil due to the number of the vehicle for shipment that is shown in Eq. (16).

\sum_{v} Y_{v'k't} \leq PO2 \times cap_{k't}^{v'} \quad \forall k', t \tag{16}

Since products type two are produced by used engine oil, the quantity of production is specified using constraint (17).

\sum_{d} WEId_{d}Q_{djt} = \sum_{k'} PO2 \times X_{v'jt}^{v'} \quad \forall j, t \tag{17}

Constraints (18)–(25) are defined to control variables to be activated when a center is open. If a center is open, the right side of the constraint can take amount.

\begin{align*}
X_{pct} & \leq M \times DES_{e} \quad \forall r, p, e, t \tag{18} \\
X_{pdt} & \leq M \times HES_{h} \quad \forall r, p, h, t \tag{19} \\
Y_{vct} & \leq M \times DES_{e} \quad \forall r, e, v, t \tag{20} \\
Y_{vht} & \leq M \times HES_{h} \quad \forall r, h, v, t \tag{21} \\
Y_{vct}' & \leq M \times CES_{f} \quad \forall v, c, f \tag{22} \\
Y_{vht}' & \leq M \times HES_{h} \quad \forall v, h, t \tag{23} \\
X_{fjt}' & \leq M \times CES_{f} \quad \forall f, j, t \tag{24} \\
X_{hjt}' & \leq M \times HES_{h} \quad \forall h, j, t \tag{25}
\end{align*}

The domain of decision variables is specified in Eq. (26).

\begin{align*}
DES_{e} & \in \{0, 1\} \quad \forall e \in E \\
CES_{f} & \in \{0, 1\} \quad \forall f \in F \\
HES_{h} & \in \{0, 1\} \quad \forall h \in H \\
Q_{pct} & \geq 0, \text{ integer} \quad \forall r, p, t \\
X_{pdt} & \geq 0, \text{ integer} \quad \forall r, p, k, t \tag{26} \\
Y_{vct} & \geq 0, \text{ integer} \quad \forall r, k, v, t \\
L_{vct} & \geq 0, \text{ integer} \quad \forall r, v, t \\
Y_{v'k't}, X_{v'k't} & \geq 0 \quad \forall k', v, t
\end{align*}

Solution methodology

Different solution methods are utilized in order to solve problems. In case of simple problems, routine methods can be used. However, by increasing the problem size, the routine solution methods would not work efficiently (Leu and Yang 1999). In these cases, using optimization techniques that are called evolutionary techniques are advantageous. Genetic algorithm (GA), particle swarm optimization (PSO), and simulated annealing (SA) are of these methods.

In this paper, three meta-heuristic algorithms are utilized for optimization. Using different algorithms can validate the accuracy of results. Since the model involves three objective functions, using multi-objective solution methodology is necessary. Non-dominated sorting genetic algorithm (NSGA-II), multi-objective simulated annealing (MOSA), and multi-objective particle swarm optimization (MOPSO) are used for this purpose. Two encoding schemes are proposed which are priority-based and permutation encoding. In the priority-based representation, the priorities of the nodes are shown by the values in a list. According to this method, the nodes with higher values of priorities are selected earlier (Peng and Wei 2008; Taleizadeh et al. 2010). In this paper, priority-based encoding is utilized due to its compatibility with the problem.

Chromosome representation

Chromosomes are encoded using three methods: (1) encoding based on edge, (2) encoding based on vertex, and (3) encoding based on edge-and-vertex (Gen and Cheng 2000). Using the last method, sources and depots are used. When |K| is the number of sources and |J| is the number of depots, the dimension of the proposed matrix for GA representation will be |K|\times|J| and the length of this matrix is |K| + |J|. The values of genes for this chromosome are between 1 and |K| + |J|. Each gene presents two kinds of values that is the locus and the allele. The locus shows the position of the value that is representative of the source or depot and the allele which is the value of the priority. Since transportation trees are generated for a set of sources and depots, the highest priority should be considered in each chromosome to select a source or depot (Altiparmak et al. 2009). These transportation problems should be solved to obtain the flow between a source and a depot. As the constrained optimization problems require constraints to be met, feasibility check is necessary. To prevent from generating infeasible solutions, using the repair mechanism is mandatory. Priority-based representation method is used to generate feasible solutions without using excessive repair mechanism (Gen and Cheng 2000). To apply priority-based encoding, stages should be clarified in the SC. The proposed SC is divided into four stages which is shown in Fig. 2. These stages are used in all the proposed algorithms such as NSGA-II, MOPSO and MOSA.

NSGA-II

NSGA-II is a multi-objective optimization algorithm which is used to generate Pareto solutions for multi-objective models. The main procedure of NSGA-II is based on the GA that is shown in Fig. 13. Generated populations are
ranked using non-dominated sorting procedure (Pasandideh et al. 2015; Safaei et al. 2017). Ranked solutions create the Pareto fronts. The crowding distance and the rank are calculated for each solution. New solutions generated by crossover and mutation operators are added to the current populations. New population is sorted and the population size is selected from these solutions (Pasandideh et al. 2015; Taleizadeh et al. 2011). Populations consist of so-called chromosomes that are the strings of values called genes. According to this definition, a chromosome is an initial solution. A chromosome is designed for each stage. The representation for each stage is shown in Fig. 3. The encoding procedure that is specified for each stage is presented in the “Appendix”.

Crossover and mutation operators

Due to defining priorities in these chromosomes, mutation and crossover should be specified in a way that duplication will be prevented. To perform crossover, order crossover (OX) was implemented. Two cut points are determined in each chromosome. The first cut point is determined in each row and the second one is determined in each column. As shown in Fig. 4, the darker parts are selected from the first parent and the white part is selected from the second one. The rest of priorities in each row are obtained from the second parent (Pasandideh et al. 2015).

In case of mutation, two positions are changed randomly for all the periods. The mutation is shown in Fig. 5.
MOSA

Simulated annealing (SA) is adapted from metal annealing. In this process, the metal is heated so that its particle can move freely. As the temperature decreases, the movement will be more limited. Solutions in problems are defined as particles in the metal. A minimum temperature is defined in simulated annealing process to which the initial temperature is decreased (Bandyopadhyay et al. 2008). In higher temperatures, solutions can change more freely in comparison to lower temperatures. SA provides the acceptance of worse solutions with a certain probability which increases the domain of local search (Santosa and Kresna 2015). Simulated annealing generates new solutions in the neighborhood of the original solution (Altiparmak et al. 2009). Since SA is a single-objective algorithm, MOSA provides a method for generating Pareto solutions. In this method, the concept of domination is used for ranking the populations. Solutions are allocated to different fronts so that the solutions in each front are non-dominated. Solutions are selected according to their ranks (Sarrafha et al. 2015). The pseudo code of simulated annealing is represented below. SA uses local search for finding solutions. A solution is generated at random initially which is updated by neighborhood procedure. If the objective function is reduced, the current solution is changed otherwise the solutions will be retained. The iteration continues until no significant change will occur (Eglese 1990).

Procedure: Simulated annealing

Begin
Initialize and evaluate solution;
Set the initial and final temperature $T_0, T_{\text{final}} > 0$;
While ($T > T_{\text{final}}$) do
Sub-iteration $i$ & neighbor $j$
New solution= Create neighbor;
Calculate $\delta = f(\text{New solution}) - f(\text{solution})$;
If $\delta > 0$ then solution=New solution
Else if random $(0, 1) < \exp(-\delta / T)$ then solution=New solution;
Update $T$;
End

MOPSO

Particle swarm optimization (PSO) works with particles that is originally taken from the bird and fish movements (James and Russell 1995). PSO searches the solution space using particles that are updated in each iteration. Each particle is specified by its position and velocity. A swarm of particles is initialized randomly which will be modified according to two experiences: the local and the global best. The local best (personal best ($P_{\text{best}}$)) is the best experience for each particle that is ever found and the global best ($G_{\text{best}}$) is the best position among all the particles. Considering these two best sets, the velocity is computed for each particle by subtracting the local and global best from the current position (Prasanna Venkatesan and Kumanan 2012). The mechanism of updating each particle is defined by changing the velocity randomly.

Velocity update

Although the priority-based representation simplifies the optimization procedure, updating velocities may be too complex because the routine velocity formula cannot update the permutation. In formula (27), $c_1$ and $c_2$ are the local and global learning coefficients.

$$v'_{ij} = v_{ij}^{t-1} + c_1 r_1 (p_{ij}^{t-1} - x_{ij}^{t-1}) + c_2 r_2 (G_{ij}^{t-1} - x_{ij}^{t-1}),$$

In this study, a specific mechanism for updating the priority-based encoding approach is used which keeps the priority values exclusive. In order to create such a mechanism, the theory of vector similarity is defined. According to this theory, the solutions are considered as vectors. The vector resemblance-degree between the current and Pbest position ($p_i$) and the current and the Gbest position ($g$) specifies the way
which that the current vectors \((x_i)\) should be changed in order to obtain velocity. The vector of resemblance-degree between the current particle (vector) and Pbest is denoted by \(r(p_i, x_i)\) and similarly Gbest by \(r(g, x_i)\). Vectors are specified by their magnitudes and directions. The magnitude of a vector is calculated using formula (28).

\[
\|x_i\| = \sqrt{\sum_{j=1}^{k} x_{ij}^2} = \sqrt{\sum_{j=1}^{k} j^2}, \quad i = 1, 2, \ldots, k
\]  

(28)

In order to calculate the vector resemblance-degree, the direction cosine between \(x_i\) and Pbest or Gbest vector is calculated.

\[
m(p_i, x_i) = \frac{\sum_{j=1}^{k} x_{ij} * p_{ij}}{\sqrt{\left(\sum_{j=1}^{k} x_{ij}^2\right) * \left(\sum_{j=1}^{k} p_{ij}^2\right)}},
\]

\[
m(g, x_i) = \frac{\sum_{j=1}^{k} x_{ij} * g_j}{\sqrt{\left(\sum_{j=1}^{k} x_{ij}^2\right) * \left(\sum_{j=1}^{k} g_j^2\right)}}
\]

(29)

Formula (30) is used in order to specify the number of dimensions to be changed. In this formulation, “\(r\)” is a random number within \([0, 1]\) and “\(w\)” is inertia weight parameter.

\[
s^{t+1}_i = (r, w)s^t_i + (1 - r, w)\left(\frac{1 - m(p_i, x^t_i)}{2} + \frac{1 - m(g, x^t_i)}{2}\right)
\]

(30)

After a change occurred in a vector, velocity should be updated. The following formula calculates new velocity:

\[
v^{t+1}_i = w * v^t_i + c_1r(p_i - x^t_i) + c_2r(g - x^t_i)
\]

(31)

**Position update**

In particle swarm optimization, positions are updated considering velocities. In this study, positions take an integer value. Indeed, a permutation of integer numbers makes a string which creates the solutions. Since integer numbers compose the positions, velocities should also be an integer. The exclusive updating procedure for permutation string is presented in (32).

1. Round the velocity and number of dimensions.

\[
v^{t+1}_{ik} = \text{int}(v^{t+1}_{ik}), \quad s^{t+1}_{ik} = \text{int}(s^{t+1}_{ik})
\]

(32)

2. New positions of a vector are calculated according to the following procedure. Formula (33) generates values in interval \([1, n]\). When a value in the permutation changes from \(l_1\) to \(l_2\), another change should be done to keep the values in permutation exclusive. This change is replacing \(l_2\) by \(l_1\).

\[
x^{t+1}_{ik} = \begin{cases} x^{t}_{ik} + \frac{v^{t+1}_{ik}}{n}, & v^{t+1}_{ik} \geq 0 \\ x^{t}_{ik} + \frac{v^{t+1}_{ik}}{n} + n, & v^{t+1}_{ik} < 0 \end{cases}
\]

(33)

The graphical presentation of the permutation position updating is shown in Fig. 6 that can be useful for better understanding of this procedure (Peng and Wei 2008).

**Fig. 6** Position update
Performance measure

There are some measures defined for evaluating multi-objective meta-heuristics algorithms. In this study, four performance metrics are considered for assessing the proposed algorithms.

Number of Pareto solutions (NPS): This measure presents the number of Pareto optimal solutions in each algorithm.

Mean ideal distance (MID): This measure examines how close the Pareto solutions and ideal point \((0, 0)\) are.

\[
MID = \frac{\sum_{i=1}^{n} c_i}{n}
\]

\[
c_i = \sqrt{f_{1i}^2 + f_{2i}^2 + f_{3i}^2}
\]

First objective function

Second objective function

Third objective function

Spread of non-dominance solution: This measure shows the diversity of Pareto solutions. The more diverse are Pareto solutions, the higher is this measure. The last metric is CPU time that measures running time of each algorithm (Karimi et al. 2010).

\[
SNS = \sqrt{\frac{\left(\sum_{i=1}^{n} (MID - c_i)\right)^2}{n - 1}}
\]

Parameter settings

Optimization algorithms operate using different parameters, e.g., GA uses crossover and mutation rates, number of iterations and populations, selection pressure and other parameters. Parameter setting is necessary in order to run algorithms in its best performance. In addition, it decreases the number of experiments and CPU time (Sarrafha et al. 2015). There are several parameter tuning approaches for meta-heuristics algorithms referring to the historical researches, try and error, arranging full factorial experiments, Taguchi method, response level, neural network, and meta-heuristic algorithms (prior or during the main optimization algorithm). Taguchi presents the optimized number of controllable factors. In this study, Taguchi method is utilized because it is based on the design of experiment (DOE). The purpose of the design of experiments is performing a set of tests that make meaningful changes in input variables to evaluate the impact of response variables. The response variable is obtained at the end of each experiment. A number of experiments are obtained considering how many factors and levels are defined. The total number of experiments is calculated using the formula (36).

| Algorithm | Level 1 | Level 2 | Level 3 |
|-----------|---------|---------|---------|
| NSGA-II   | 15      | 25      | 50      |
| Rate of crossover | 0.7 | 0.8 | 0.95 |
| Rate of mutation | 0.2 | 0.25 | 0.3 |
| Iteration | 50      | 80      | 100     |
| MOSA      | 5       | 10      | 15      |
| Neighbor | 3       | 8       | 13      |
| Initial temperature | 95 | 200 | 350 |
| Final temperature | 5 | 10 | 15 |
| Swap probability | 0.4 | 0.45 | 0.5 |
| Reversion probability | 0.6 | 0.55 | 0.5 |
| Sub-iteration | 5 | 25 | 50 |
| Iteration | 50      | 80      | 120     |
| MOPSO     | 0.7     | 0.85    | 0.95    |
| Local learning coefficient | 0.95 | 0.85 | 0.75 |
| Global learning coefficient | 0.25 | 0.5 | 0.75 |
| Inertia weight | 5 | 10 | 15 |
| Grid number | 50 | 80 | 100 |
| Population | 10      | 15      | 20      |
| Repository | 5       | 10      | 15      |

Total number of experiments

\[
= \text{(number of levels)}^{\text{number of factors}} \times \text{number of iterations}
\]

Two strategies are defined for performing the experiments: (1) changing one factor against other fixed factors. This method is called “one factor at time (OFAT).” (2) Changing some factors simultaneously, so that the impact of each factor and their interactions will be specified. The second one is the so-called “DOE”.

To simplify the computation, orthogonal arrays are designed which presents a number of experiments considering a number of factors and levels. In order to perform Taguchi analyses, two methods are recommended. First, a standardized method in which analysis of variance (ANOVA) is used. Second, using signal to noise (S/N) ratio for the same stages in the analysis. The higher this ratio is, the variance around a specific amount will be lower (Roy 2010). Three levels are defined for each parameter in each algorithm according to its characteristics that is shown in Table 4.

Ten random instances are given to compare proposed algorithms. Table 5 shows dimensions of the instances. Parameters are generated randomly according to parameter ranges defined in Table 6.

Number of runs were specified by the orthogonal arrays considering number of factors and levels. An L9 design was recommended for NSGA-II, L27 design for MOSA.
and L27 for MOPSO algorithm according to orthogonal arrays in Minitab software. Since ten instances were generated, 270 test problems were solved for MOSA algorithm, 90 test problems were solved for NSGA-II and 270 test problems for MOPSO. Therefore, 630 runs were totally solved for Taguchi analysis. The outputs were normalized using formula (37) which is shown in Table 7. Table 7 contains three measures of MID, NPS, and SNS. As observed, the mechanism that is used for position update of MOPSO algorithm provides a spread search of the solution space. Thus, this procedure is very useful for a wide search.

\[
N = \frac{|X - Y|}{Y} \quad X : \text{Present solution} \\
\quad Y : \text{Best solution} 
\]

(37)

The formula (38) was utilized for calculating the scores. Since all results are normalized, test problems with minimum scores are selected.

\[
\text{Score} = \sum_{i=1}^{I} w_i v_i \quad w_i : \text{weight of criterion} \\
\quad v_i : \text{value of data} 
\]

(38)

The computations are executed on a Desktop Computer with core™ i7 3.50 GHz CPU. The scores and CPU time are computed for each algorithm, shown in Table 8. In order to compare three algorithms, performance measures “w” consisting of weighted summation of MID, NPS and SNS and CPU time were used. Both criteria of NSGA-II are minimum, so it is ranked first. Figures 7, 8, 9, and 10 show the comparison among three algorithms considering performance measures.

According to the analyses, NSGA-II is selected. Taguchi approach is conducted for parameter tuning which can enhance the performance of NSGA-II.
analyses are performed for instances that are generated using data in Table 5. In this table, instances are classified to three types of problems: small, medium and big-size problems. Three instances are selected from Table 5, i.e. 1, 5 and 10, on behalf of all the problems. Pareto fronts of the optimal solutions for instances 1, 5 and 10 in Fig. 11 present proper diversity of the generated solutions for NSGA-II, MOSA and MOPSO.

In order to interpret results, two criteria of the mean of means and signal to noise ratio (S/N) are defined. The lower values of the mean of means and the higher S/N ratio show the better results that is shown in Fig. 12. Based on this figure, NSGA-II parameters are tuned for each instance. The tuned parameters are population, crossover rate, mutation rate, and iteration.

Conclusion and future research

In this study, a CLSC of the used engine oil is optimized using different meta-heuristics. Optimizing small-size problems can be performed using optimization software. However, in case the dimensions of the problems increase, the new optimization tools dominate the routine optimization software. The Meta-heuristic algorithms are one of these methods that assists engineers to solve the big-size problems faster. In this paper, three meta-heuristic algorithms are utilized to optimize the proposed mathematical model. Since a multi-objective model is discussed, a multi-objective meta-heuristic should be applied in this case. The NSGA-II, MOSA and MOPSO are three algorithms implemented to deal with the multi-objective model. The feasibility control of the constraints is very essential, so it encouraged us to concentrate on the priority-based encoding and position update very carefully. In all the algorithms, a priority-based encoding is used. Instances are generated randomly to compare the capability of three algorithms. To compare three algorithms, MID, NPS, SNS, and CPU time are used as measures. Considering these criteria, NSGA-II outperformed other algorithms for the closed-loop supply chain of engine oil. This algorithm works better than other proposed algorithms for sophisticated problems such as closed-loop used engine oil logistics. Taguchi approach that is used in this study helps enhance the performance of the present algorithm and determine the best present of operational parameters.

This study could be pursued by the direction of solution methodology and parameter settings. Other
algorithms can be provided to compare the performance of the proposed algorithms and other algorithms. For enthusiasts of supply chain management, there would be an option to modify the mathematical model to cover the requirement in used engine oil collection. Vehicle routing for distribution and collection part of the proposed supply chain can be complementary to the proposed mathematical model. In addition, other objective functions could be added to the present objective functions to cover real-world problems. Considering the uncertainty of real-world problems can be helpful for this purpose.

Fig. 11 Optimal solutions for instances 1, 5 and 10
Fig. 12  Mean of means and S/N ratio of instances 1, 5 and 10

**Appendix**

See Figs. 13, 14, 15, 16, 17, 18, and 19.
Fig. 13 Procedure of the NSGA-II

Fig. 14 Chromosome representation for stage 2

Fig. 15 Chromosome representation for stage 3

Fig. 16 Chromosome representation for stage 4
Fig. 17 Swap

Fig. 18 Reversion

Fig. 19 Insertion
Procedure of priority-based decoding for stage 1

\( V \): Set of vendors, \( C \): Set of collection centers, \( H \): Set of hybrid centers, \( T \): set of periods,
\( O = \{ C, H \} \): set of collection and hybrid centers;
\( tv_{vc} \): Transportation cost from vendor \( v \) to collection center \( c \) for period \( t \), \( \forall v, c, \forall t \),
\( tv_{ah} \): Transportation cost from vendor \( v \) to hybrid center \( h \) for period \( t \), \( \forall v, \forall h, \forall t \),

Inputs:
\( tv_{o\cdot} \): \{tv_v, tv_h\},
\( cap_{o\cdot} \): Capacity of distribution center \( o \) for period \( t \), \( \forall o, \forall t \),
\( cap_v \): Capacity of vendor \( v \) for period \( t \), \( \forall v, \forall t \),
\( ch(t \times (v+c+h)) \): chromosome, \( \forall v \in V, c \in C, h \in H, t \in T \).

Outputs:
\( Y_{C_v} \): The amount of used oil transported from vendor \( v \) to collection center \( o \) in period \( t \).

For \( t = 1 \) to \( T \)

While \( \forall \{ ch(\cdot) \} \neq 0 \):

Step 1. Chromosome generation
\( ran \leftarrow \arg \max \{ ch(u), u \in v + |c| + |h| \}; \)

Step 2. Determining sources and depots
If \( ran \leq v \),
\( v^* \leftarrow ran \), a vendor is selected
\( o^* \leftarrow \arg \min \{ tv_{o\cdot} \} \), selecting a collection center with minimum transportation cost
else
\( o^* \leftarrow ran \), a collection center is selected
\( v^* \leftarrow \arg \min \{ tv_{v\cdot} \} \), selecting a vendor with minimum transportation cost

Step 3. \( Y_{C_{v^*o^*t}} = \min \{ cap_{o\cdot}, cap_v \} \), assigning from capacity to shipment between vendor and collection center

Step 4. Updating capacities
\( cap_{o\cdot} = cap_{o\cdot} - Y_{C_{v^*o^*t}} \), \( cap_v = cap_v - Y_{C_{v^*o^*t}} \)

End

Step 5. Determining which distribution center(s) to be opened
If \( \sum(Y_{C_{v^*o^*t}} (:, o^*,:)) \geq 0 \) and \( o^* \leq c \)
\( CC_{o^*} = 1 \)
If \( \sum(Y_{C_{v^*o^*t}} (:, o^*:)) \geq 0 \) and \( o^* > c \)
\( HC_{o^*} = 1 \)
Procedure of priority-based decoding for stage 2

Inputs:
C: Set of collection centers, H: Set of hybrid centers, r: set of manufacturing type two,
T: set of periods, O = C ∪ H: set of manufacturing centers,
\text{tow}_{o,w}: Transportation cost from collection center to manufacturing center for period t, ∀o, ∀w, ∀t,
\text{cap}_{k,w}: Capacity of producing product k in manufacturing center w for period t, ∀k, ∀w, ∀t,
\text{Y}_{v,o,t}: amount of used oil transported from vendor v to collection center o for period t, ∀v, ∀o, ∀t,
\text{ch}(t × (c+h+r)): chromosome, c ∈ C, h ∈ H, r ∈ W, t ∈ T

Outputs: \text{XC}_{o,w,t}: The amount of used oil transported from collection center o to manufacturing w in period t

For t = 1 to T
While all \{ch(t)\} ≠ 0
Step 1. Chromosome generation
\text{ran} ← \arg\max \{ch(u), u ∈ |c| + |h| + |r|\};
Step 2. Determining sources and depots
\text{ran} ≥ o, o^* ← \text{ran}, a collection center is selected
r^* = \arg\min\{\text{tow}_{o,w} | \text{ch}(t,r) ≠ 0, ∀o ∈ O\}, selecting a manufacturing type two with minimum transportation cost
else
r^* ← \text{ran}, a manufacturing type two is selected
o^* = \arg\min\{\text{tow}_{o,r} | \text{ch}(t,o) ≠ 0, ∀r ∈ W\}, selecting a collection center with minimum transportation cost
Step 3.
\text{A} = \sum_{v} Y_{v,o,t}, \text{B} = \sum_{k} \text{weight}_{k} \times \text{teta}_{k} \times \text{cap}_{k,r,t}
\text{XC}_{o,r,t} = \min\{A, B\}, assigning from capacity to shipment between collection center and manufacturing type two
Step 4. Updating capacities :
A = A - \text{XC}_{o,r,t}, B = B - \text{XC}_{o,r,t}
End
End
Procedure of priority-based decoding for stage 3

Inputs:
\[ D \text{: Set of distribution centers, } H \text{: Set of hybrid centers}, m \text{: set of manufacturing type one, } r \text{: set of manufacturing type two, } \]
\[ T \text{: set of periods, } O = \{D, H\} \text{: set of distribution and hybrid centers, } W = \{m, r\} \text{: Set of manufacturing centers,} \]
\[ t_{o_{kwot}} \text{: Transportation cost of product } k \text{ from manufacturing center to distribution center for period } t, \forall k, \forall w, \forall o, \forall t, \]
\[ c_{w_{kwt}} \text{: Capacity of producing product } k \text{ in manufacturing center } w \text{ for period } t, \forall k, \forall w, \forall t, \]
\[ Y_{D_{kost}} \text{: amount of product } k \text{ transported from distribution } o \text{ to vendor } v \text{ for period } t, \forall k, \forall o, \forall v, \forall t, \]
\[ X_{C_{ort}} \text{: amount of used oil transported from collection center } o \text{ to manufacturing center } r \text{ for period } t, \forall o, \forall r, \forall t, \]
\[ \text{weight}_k \text{: weight (kg) of product } k, \text{ tet}_k \text{: amount of oil needed for producing each product} \]
\[ ch(t \times k \times (m+r+d+h)) \text{: chromosome, } k \in K, m \in W, r \in W, d \in D, h \in H, t \in T \]

Outputs:
\[ X_{D_{kwo}} \text{ : The amount of product } k \text{ transported from manufacturing } w \text{ to distribution } o \text{ in period } t \]

For \( t = 1 \) to \( T \)
While all \( \{ch()\} \neq 0 \)

Step 1. Chromosome generation:
\[ \text{ran} \leftarrow \arg \max \left\{ch(u), u \in \left\{k | |(m | + | r | + | d | + | h |)\right\}\right\}; \]

Step 2. Selecting product:
\[ k^* = \left[\frac{\text{ran}}{m | + | r | + | d | + | h |}\right]; \]

Step 3. Determining sources and depots
\[ a = \text{ran} - (k^* - 1) \times (m | + | r | + | d | + | h |), \]
If \( a \leq |m | + | r | \),
\[ w^* \leftarrow a, \text{ a manufacturing center is selected} \]
Else
\[ o^* = \arg \min \{t_{o_{kwot}} | ch(t,w) \neq 0, \forall w \in W\}, \text{ selecting a distribution center with minimum transportation cost} \]
\[ w^* = \arg \min \{t_{o_{kwot}} | ch(t,w) \neq 0, \forall o \in O\}, \text{ selecting a manufacturing center with minimum transportation cost} \]

If \( k^* \leq i \) go to step 4 otherwise go to step 6
\[ A = \sum_{w} X_{D_{k_{w't}}}, B = cap_{k_{w't}} \]

Step 4. Determining amount of shipment from manufacturing center to distribution center
\[ X_{D_{k_{wo't}}} = \min \{A, B\}, \]

Step 5. Updating capacities:
\[ A = A - X_{D_{k_{wo't}}}, B = B - X_{D_{k_{wo't}}}, \]
Else

Step 6. Determining amount of shipment from manufacturing center to distribution center
\[ A' = \sum_{o} X_{C_{ort}} A = \sum_{v} Y_{D_{k_{ov't}}} \]
\[ X_{D_{k_{wo't}}} = \min \{A', \text{cap}_{k_{w't}}, A\}, \]

Step 7. Updating capacities
\[ A' = A' - X_{D_{k_{wo't}}}, \text{weight}_k \times \text{tet}_k \times X_{D_{k_{wo't}}}, B = B - X_{D_{k_{wo't}}}, \text{ cap}_{k_{w't}} = \text{cap}_{k_{w't}} - X_{D_{k_{wo't}}} \]

End
End
Procedure of priority-based decoding for stage 4

Inputs:
D: Set of distribution centers, H: Set of hybrid centers, V: set of vendors, T: set of periods,
O = [D,H]: set of distribution and hybrid centers,
tovkvt: Transportation cost of product k from distribution center to vendor for period t, ∀k, ∀o, ∀v, ∀t,
capot: Capacity of producing product k in manufacturing center w for period t, ∀k, ∀w, ∀t,
Dkvt: Demand of vendor v for product k for period t, ∀k, ∀v, ∀t,
weightk: weight (kg) of product k.

Outputs: YDkotv: The amount of product k transported from distribution o to vendor v in period t
For t=1 to T
While all {ch(·)} ≠ 0
Step1. Chromosome generation:
ran ← arg max {ch(u), u ∈ [k | ×(|d | +| h | +| v |)]};
Step2. Selecting product:
k* = \left\lfloor \frac{\text{ran}}{|d | +| h | +| v |} \right\rfloor .
Step3. Determining sources and depots
D = ran − (k*−1) ×(|m | +| r | +| d | +| h |),
If a ≤ |d | +| h |, a distribution center is selected
a* ← a, a vendor is selected

Step 4. Determining amount of shipment from distribution center to vendor
YDkotv = \min \{\text{cap}^\prime_{ov}, D_{kvt}\},
Step 5. Updating capacities
\text{cap}^\prime_{ot} = \text{cap}^\prime_{ot} − YD_{kotv}, \quad D_{kvt} = D_{kvt} − YD_{kotv},
End
End

References

Altiparmak F, Gen M, Lin L, Karaoglan I (2009) A steady-state genetic algorithm for multi-product supply chain network design. Comput Ind Eng 56:521–537
Bandyopadhyay S, Saha S, Maulik U, Deb K (2008) A simulated annealing-based multiobjective optimization algorithm: AMOSA. IEEE Trans Evol Comput 12:269–283
Chen Y-W, Wang L-C, Wang A, Chen T-L (2017) A particle swarm approach for optimizing a multi-stage closed loop supply chain for the solar cell industry. Robot Compt Integr Manuf 43:111–123
Eglese R (1990) Simulated annealing: a tool for operational research. Eur J Oper Res 46:271–281
Fahimnia B, Farahani RZ, Marian R, Luong L (2013) A review and critique on integrated production–distribution planning models and techniques. J Manuf Syst 32:1–19
Gen M, Cheng R (2000) Genetic algorithms and engineering optimization, vol 7. Wiley, New York
Ghasimi SA, Ramli R, Saibani N (2014) A genetic algorithm for optimizing defective goods supply chain costs using JIT logistics and each-cycle lengths. Appl Math Model 38:1534–1547
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(1):371–384
James K, Russell E (1995) Particle swarm optimization. In: Proceedings of 1995 IEEE international conference on neural networks, pp 1942–1948
Jolai F, Razmi J, Rostami N (2011) A fuzzy goal programming and meta heuristic algorithms for solving integrated production: distribution planning problem. CEJOR 19:547–569
Karimi N, Zandieh M, Karamooz HR (2010) Bi-objective group scheduling in hybrid flexible flowshop: a multi-phase approach. Expert Syst Appl 37:4024–4032
