

Behmanesh, Elham; Pannek, Jürgen

**Article — Published Version**

## Taguchi Analysis for Improving Optimization of Integrated Forward/Reverse Logistics

Journal of the Operations Research Society of China

**Provided in Cooperation with:**

Springer Nature

*Suggested Citation:* Behmanesh, Elham; Pannek, Jürgen (2022) : Taguchi Analysis for Improving Optimization of Integrated Forward/Reverse Logistics, Journal of the Operations Research Society of China, ISSN 2194-6698, Springer, Berlin, Heidelberg, Vol. 11, Iss. 3, pp. 529-552, <https://doi.org/10.1007/s40305-021-00380-7>

This Version is available at:

<https://hdl.handle.net/10419/309272>

**Standard-Nutzungsbedingungen:**

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

**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/>



# Taguchi Analysis for Improving Optimization of Integrated Forward/Reverse Logistics

Elham Behmanesh<sup>1</sup> · Jürgen Pannek<sup>2,3</sup>

Received: 24 May 2019 / Revised: 4 October 2021 / Accepted: 2 November 2021 /

Published online: 28 January 2022

© The Author(s) 2022

## Abstract

The distribution–allocation problem is known as one of the most comprehensive strategic decisions. In real-world cases, it is impossible to solve a distribution–allocation problem completely in acceptable time. This forces the researchers to develop efficient heuristic techniques for the large-term operation of the whole supply chain. These techniques provide near optimal solution and are comparably fast particularly for large-scale test problems. This paper presents an integrated supply chain model which is flexible in the delivery path. As solution methodology, we apply a memetic algorithm with a novelty in population presentation. To identify the optimum operating condition of the proposed memetic algorithm, Taguchi method is adopted. In this study, four factors, namely population size, crossover rate, local search iteration and number of iteration, are considered. Determining the best level of the considered parameters is the outlook of this research.

**Keywords** Integrated logistics network · Flexible path · Memetic algorithm · Taguchi analysis

**Mathematics Subject Classification** 90B06 · 68W50 · 91G70

## 1 Introduction

Supply chain networks describe the flow and movement of goods by linking several facilities such as plants, distributions, and retailers in forward flow[1,2] and collec-

---

✉ Elham Behmanesh  
beh@biba.uni-bremen.de

Jürgen Pannek  
juergen.pannek@googlemail.com

<sup>1</sup> International Graduate School for Dynamics in Logistics, University of Bremen, Bremen, Germany

<sup>2</sup> Faculty of Production Engineering, University of Bremen, Bremen, Germany

<sup>3</sup> BIBA Bremer Institut für Produktion und Logistik GmbH, Bremen, Germany

tion/inspection centers and disposal centers in backward flow [3]. Supply chain goals are summarized in minimization of total cost [4,5] or maximization of profit [6] according to the interested outcome. Supply chain activities involve determining the optimal number and capacity of facilities as well as the flow between them [5]. Network configuration comes first in any supply chain network design and needs to be optimized for a long-lasting efficient operation of the entire supply chain. Therefore, long-term strategic decisions take priority over tactical and operational levels. A result of a comprehensive review of the supply chain literature by Thomas and Griffin [7] shows that distribution cost expenditures are equal to about 30 percent of product costs. Hence, focusing on decreasing transportation cost has a significant impact on reducing the distribution cost of the supply chain network. In this regard, considering flexibility in delivery path to have optional short ways delivery is noticeable. Besides, fast and on-time delivery of products plays an important role in customer satisfaction [8,9]. Although different delivery paths improve efficiency, it reveals the problem more complex. But the problem is, the purpose of reducing delivery time is often conflict with the goal of reducing logistics cost [10]. To deal with the issue of cost efficiency and network responsiveness, researchers have proposed models to optimize both but results are typically limited to shipments between consecutive stages or just indirect shipment mechanisms [8,9,11]. The full capacitated graph in forward flow considered in this study allows us to solve conflicting goals of profit and responsiveness which otherwise may lead to greater cost. As environmental protection forced firms to pay more attention to collect, recover, recycle and safe disposal in a supply chain network [12], reverse distribution needs to be added in any supply chain network. While the reverse activity could be useful for environmental protection, industries can use returned product for economic benefits [13]. Although industrial players are forced to handle returned products, most of logistics networks are not equipped to deal with this requirement [12]. Therefore, management of product return flow is becoming an essential part of each supply chain. Within this paper, we attempt to include the reverse flow through an integrated design of forward/reverse supply chain network. The proposed integrated design leads us to a closed-loop supply chain network which is avoid sub-optimal solutions derived by separated design [14], cf. Fig. 1 for a sketch.

As shown in Fig. 1, we have raw materials from supplier to plant. New products are shipped from plant to customer through distribution center and retailer. Three different delivery paths are considered in the forward flow to enrich the model in order to being close to customer, reducing transportation cost and increasing customer satisfaction. Except normal delivery, which is from any stage to another adjoining one, we added direct shipment and direct delivery. In direct delivery, goods are transported from distribution centers to customers by skipping retailers or from plants to retailers by skipping distribution centers. In direct shipment, products are transferred from plants to customers directly. In the backward flow, returned products are collected by collection centers and, after inspection, the recoverable products are shipped to plants, and scraped products are transferred to disposal centers for a safe disposal.

According to the aforementioned description, in this study, we attempt to add the reverse flow through an integrated design to the presented network. Also, a full delivery graph in forward flow between plants and customers is considered to increase the performance of the supply chain network. This model can be formulated into an

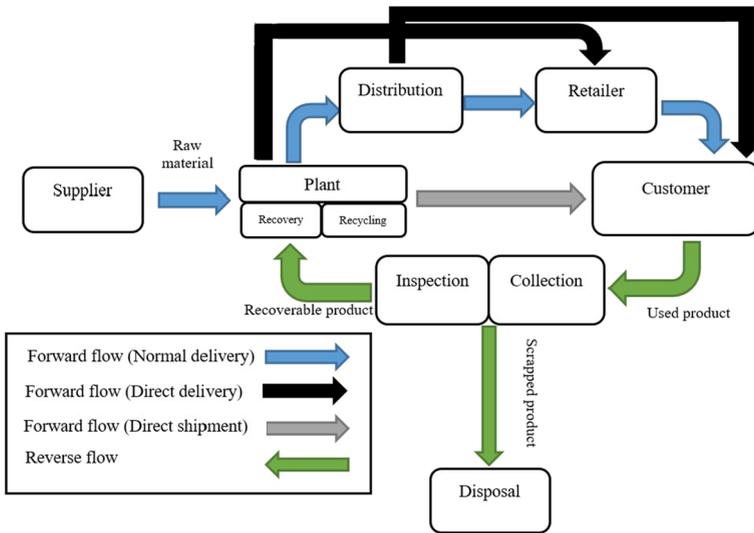


Fig. 1 Framework of the proposed closed-loop supply chain network

integer linear programming, which traditional methods fail to solve in acceptable time, particularly when we are facing with large size problems. To overcome this problem, nontraditional solutions such as heuristics and metaheuristics [8,11,15–17] have been proposed in recent years. Within this paper, we present a memetic algorithm with a new chromosome representation as well as updated operators. Each algorithm has some parameters. Some information regarding these parameters would be useful to improve the performance of the results. In this study, Taguchi method is applied to improve operation condition. Determining the best level of parameters is the main contribution of this work.

In the last several decades, growing attention is being paid to the use of Taguchi method to find the best setting of parameters involved in evolutionary algorithms. Chouhan et al. [18] introduced a novel approach that had the ability in collecting end-of-life and end-of-use products from the end-users and in reducing the overall cost of the supply chain. To assess the overall cost of the supply chain, a mixed-integer linear programming model has been formulated, while the Taguchi method was used to obtain the best combinations of the algorithm parameters.

Gümüs et al. [19] analyzed and extended the Taguchi method to tune the parameters of a memetic algorithm for cross-domain search. In their work, they investigated the degree to predict the same good parameter setting faster by using a reduced time budget and showed that it was possible to predict good combinations of parameter settings with a much reduced time budget.

Alavidoost et al. [20] proposed a fuzzy mixed-integer linear programming optimization model for an integrated supply, production, and distribution supply chain of multi-product, multi-level, multi-plant. They proposed a model aimed to optimize customer satisfaction as well as chain supply costs, simultaneously, which was solved by some popular multi-objective metaheuristic algorithms. They developed a Taguchi

method to calibrate as well as control the parameters in algorithms. Additionally, they proposed a new framework for ranking the algorithms with the use of the fuzzy VIKOR method.

A mathematical model for a fixed-charge solid transportation problem for a two-stage supply chain network is introduced in Akbari et al. [21] They presented a fixed-charge solid transportation problem in a two-stage supply chain network, employing conveyances in each stage of the supply chain. They used three metaheuristic algorithms, namely GA, EM, and CSS, in accordance with priority-based encoding, tackling this NP-hard problem, and employed the Taguchi parameter design method to generate the best solutions. They showed that the best result can be obtained by the use of CSS.

Zahraee et al. [22] used dynamic simulation and the Taguchi method to design a robust blood supply chain. To do the Taguchi method, four main controllable factors including arrival rate of donors, maximum inventory level, minimum inventory level, and blood delivery policy, and one noise factor have been chosen. Their results showed that a robust blood supply chain can be achieved, while all main factors are located at a high level except for factor B.

Vinay and Sridharan [23] developed a solution methodology using ant colony optimization (ACO) for a distribution–allocation problem in a two-stage supply chain with a fixed cost for a transportation route, while Taguchi method for robust design was adopted for finding the optimum combination of parameters of the ACO algorithm. Applying the Taguchi method led to significant reduction in the total number of experiments to be carried out for finding an optimum level of parameters, namely from 5 120 to 125, i.e., 25 experiments each with five replications.

The paper is divided into six sections, and the rest of this paper is organized as follows. In Sect. 2, we describe the proposed network and formulate the considered problem into a mixed integer linear programming. Then, after a brief explanation regarding the solution methodology in Sect. 3, parameter analysis methods are presented in Sect. 4. Section 5 provides Taguchi analysis for improving optimization of the model, and results are shown. Section 6 concludes the paper and suggests future plans.

## 2 Description for Integrated Forward/Reverse Logistics Network

In this section, we formally define the considered problem and formulate the proposed supply chain network. The model deals with the minimization of transportation and operation cost of the proposed flexible integrated forward/reverse logistics network in order to determine the optimal capacity and number of each node as well as distribution. For this purpose, the problem is stated as follows. Consider  $G = (N, E)$  as a digraph, where  $N$  is the set of nodes and  $E$  the set of edges. The cost involved in the proposed model is denoted in two types: (1) fixed cost for node  $i \in N$  by  $c_i$  and (2) unit transportation cost on edge  $(i, j) \in E$  by  $c_{ij}$ . Furthermore,  $y_i$  and  $x_{ij}$  represent decision variables, where  $y_i \in \{0, 1\}$  and  $x_{ij} \in \mathbb{N}_0$  and  $a_i$  and  $b_i$ , vectors of known coefficients for the involved constraints. These variables indicate whether a stage  $i \in N$  is used, and which quantity is transported from node  $i$  to  $j$ , respectively.

In accordance to above description, the following mixed integer linear minimization problem arises

$$\begin{aligned}
 \min_{x_{ij}, y_i} \quad & \sum_{(i,j) \in E} c_{ij} x_{ij} + \sum_{i \in N} c_i y_i \\
 \text{s.t.} \quad & \sum_{(i,j) \in E} a_i x_{ij} \leq b_i y_i, \\
 & x_{ij} \geq 0, \quad y_i \in \{0, 1\}.
 \end{aligned} \tag{1}$$

Next, we specialize this model to reflect the problem properties.

### 2.1 Assumption of the Proposed Model

The previously described flexible forward/reverse logistics network setting represents an integrated supply chain with seven echelons consisting of suppliers *S*, plants *P*, distribution centers *Dc*, retailers *R* and customers *C* in forward flow, as well as collection/inspection centers *Co* and disposal centers *Di* in reverse flow, cf. Fig. 2 for a schematic sketch. We like to point out that in accordance to Fig. 2, we consider a hybrid manufacturing–recovery–recycling facility as well as a hybrid collection–inspection facility. Establishing several facilities at the same location can decrease the price of the whole network in comparison with separated design.

To adapt problem (1), we impose the following assumptions:

- There are seven echelons: suppliers, plants, distribution centers, retailers, customers, collection/inspection centers, and disposal centers.

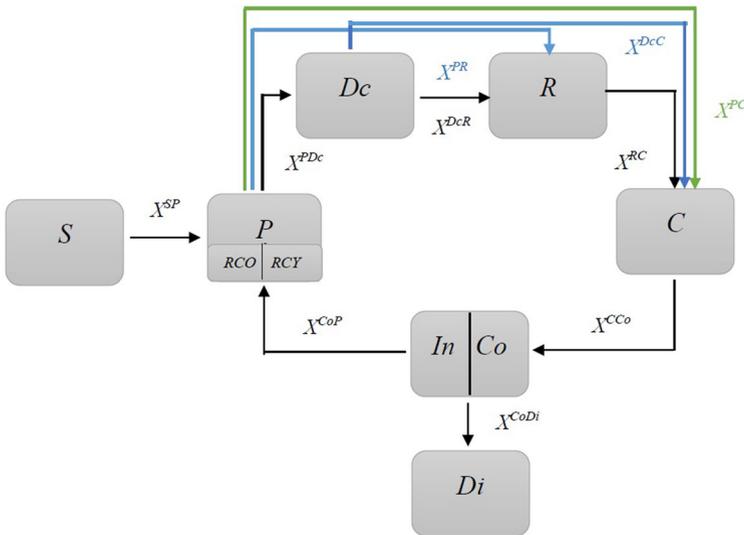


Fig. 2 Underlying structure of optimization problem

- The set of nodes is given by  $N = S \cup P \cup Dc \cup R \cup C \cup Co \cup Di$ .
- There are no edges between facilities of the same stage, the delivery graph is complete in forward flow and the return graph is simple, i.e.,  $E = (S \times P) \cup (P \times Dc) \cup (P \times R) \cup (P \times C) \cup (Dc \times R) \cup (Dc \times C) \cup (R \times C) \cup (C \times Co) \cup (Co \times Di) \cup (Co \times P)$ .
- The demands of each customer are deterministic and must be satisfied.
- The number of facilities per stage and respective capacities are limited.
- All cost parameters (fixed and variables) are known in advance.
- The transportation rates are perfect and there are no storages. Moreover, the return rate  $p_m^{\text{return}}$  as well as the disposal and recovery rates  $p_n^{\text{disposal}}$  and  $(1 - p_n^{\text{disposal}})$  are fixed. All returned products from each customer must be collected.
- The inspection cost per item for the returned products are included in the collection cost.
- The un-recyclable returned products will be sent to the disposal center. The remaining products are returned to the same plant.
- The required recycled materials are assumed to be of the same quality as the raw materials bought from suppliers and any plant chooses the raw material from the collection/inspection centers over suppliers.
- Customers have no special preference. It means, price is the same in all facilities.

**2.2 Notation**

To support the presentation of the proposed mathematical model, we first provide a verbal description of the model as follows:

$$\text{Cost} = \text{transportation costs} + \text{Fixed opening cost.} \tag{2}$$

The notation given in Tables 1, 2, 3 and 4 is used in the formulation of the MILP model.

**2.3 Mathematical Formulation**

Utilizing the notation of Tables 1, 2, 3 and 4, the structure of the MILP model can be presented as follows. The objective of this model is to minimize the total cost of the proposed supply chain. The first part of the cost function represents the variable

**Table 1** Model indicates

Indices	
$i$	Index of supplier ( $i = 1, 2, \dots, I$ )
$j$	Index of plant ( $j = 1, 2, \dots, J$ )
$k$	Index of distribution center ( $k = 1, 2, \dots, K$ )
$l$	Index of retailer ( $l = 1, 2, \dots, L$ )
$m$	Index of customer ( $m = 1, 2, \dots, M$ )
$n$	Index of collection/inspection center ( $n = 1, 2, \dots, N$ )
$o$	Index of disposal ( $o = 1, 2, \dots, O$ )

**Table 2** Model parameters

Parameters	
$I$	Number of suppliers
$J$	Number of plants
$K$	Number of distribution centers
$L$	Number of retailers
$M$	Number of customers
$N$	Number of collection/inspection centers
$O$	Number of disposal centers
$\bar{S}_i$	Amount of product supplied by supplier $i$
$\bar{P}_j$	Amount of product produced by plant $j$
$\bar{D}c_k$	Capacity of distribution center $k$
$\bar{R}_l$	Capacity of retailer $l$
$D_m$	Demand of customer $m$
$\bar{C}o_n$	Capacity of collection/inspection center $n$
$\bar{D}i_o$	Capacity of disposal center $o$
$p_m^C$	Recovery percentage of customer $m$
$P_n^{Co}$	Disposal percentage of collection/inspection center $n$

**Table 3** Model parameters

Parameters	
$c_{ij}^{SP}$	Unit cost of transportation from supplier to plant
$c_{jk}^{PDc}$	Unit cost of transportation from plant to distribution center
$c_{kl}^{DcR}$	Unit cost of transportation from distribution to retailer
$c_{lm}^{RC}$	Unit cost of transportation from retailer to customer
$c_{mn}^{CCo}$	Unit cost of transportation from customer to collection/inspection center
$c_{no}^{CoDi}$	Unit cost of transportation from collection/inspection center to disposal center
$c_{nj}^{CoP}$	Unit cost of transportation from collection/inspection center to plant
$c_{jl}^{PR}$	Unit cost of transportation from plant to retailer
$c_{km}^{DcC}$	Unit cost of transportation from distribution center to customer
$c_{jm}^{PC}$	Unit cost of transportation from plant to customer
$c_j^P$	Fixed cost of operating plant $j$
$c_k^{Dc}$	Fixed cost of operating distribution center $k$
$c_l^R$	Fixed cost of operating retailer $l$
$c_n^{Co}$	Fixed cost of operating collection/inspection center $n$
$c_n^{Di}$	Fixed cost of operating disposal center $o$

**Table 4** Model variables

Integer	
$X_{ij}^{SP}$	Amount of products transported from supplier $i$ to plant $j$
$X_{jk}^{PDc}$	Amount of products transported from plant $j$ to distribution center $k$
$X_{kl}^{DcR}$	Amount of products transported from distribution center $k$ to retailer $l$
$X_{lm}^{RC}$	Amount of products transported from retailer $l$ to customer $m$
$X_{mn}^{CCo}$	Amount of products transported from customer $m$ to coll/ins center $n$
$X_{no}^{CoDi}$	Amount of products transported from collection/inspection center $n$ to disposal center $o$
$X_{nj}^{CoP}$	Amount of products transported from collection/inspection center $n$ to plant $j$
$X_{jl}^{PR}$	Amount of products transported from plant $j$ to retailer $l$
$X_{km}^{DcC}$	Amount of products transported from distribution center $k$ to customer $m$
$X_{jm}^{PC}$	Amount of products transported from plant $j$ to customer $m$
Binary	
$X_j^P$	$\begin{cases} 1 & \text{if plant } j \text{ is used for production and remanufacturing} \\ 0 & \text{otherwise} \end{cases}$
$X_k^{Dc}$	$\begin{cases} 1 & \text{if distribution center } k \text{ is used for distribution} \\ 0 & \text{otherwise} \end{cases}$
$X_l^R$	$\begin{cases} 1 & \text{if retailer } l \text{ is used for distribution} \\ 0 & \text{otherwise} \end{cases}$
$X_n^{Co}$	$\begin{cases} 1 & \text{if collection/inspection center } n \text{ is used for collection/inspecting} \\ 0 & \text{otherwise} \end{cases}$
$X_o^{Di}$	$\begin{cases} 1 & \text{if disposal center } o \text{ is used for safe disposal} \\ 0 & \text{otherwise} \end{cases}$

costs, which are given by the direct multiplication of the transportation cost and the quantity transferred from the origin to the destination. The fixed costs of the logistics chain manufacturing facility is considered as the second part of cost function. This reveals the cost function as follows:

$$\begin{aligned}
 \min & \sum_{i=1}^I \sum_{j=1}^J c_{ij}^{SP} \cdot X_{ij}^{SP} + \sum_{j=1}^J \sum_{k=1}^K c_{jk}^{PDc} \cdot X_{jk}^{PDc} \\
 & + \sum_{k=1}^K \sum_{l=1}^L c_{kl}^{DcR} \cdot X_{kl}^{DcR} + \sum_{l=1}^L \sum_{m=1}^M c_{lm}^{RC} \cdot X_{lm}^{RC} \\
 & + \sum_{m=1}^M \sum_{n=1}^N c_{mn}^{CCo} \cdot X_{mn}^{CCo} + \sum_{n=1}^N \sum_{o=1}^O c_{no}^{CoDi} \cdot X_{no}^{CoDi} \\
 & + \sum_{n=1}^N \sum_{j=1}^J c_{nj}^{CoP} \cdot X_{nj}^{CoP} + \sum_{j=1}^J \sum_{l=1}^L c_{jl}^{PR} \cdot X_{jl}^{PR}
 \end{aligned}$$

$$\begin{aligned}
 & + \sum_{k=1}^K \sum_{m=1}^M c_{km}^{DcC} \cdot X_{km}^{DcC} + \sum_{j=1}^J \sum_{m=1}^M c_{jm}^{PC} \cdot X_{jm}^{PC} \\
 & + \sum_{j=1}^J c_j^P \cdot X_j^P + \sum_{k=1}^K c_k^{Dc} \cdot X_k^{Dc} + \sum_{l=1}^L c_l^R \cdot X_l^R + \sum_{n=1}^N c_n^{Co} \cdot X_n^{Co} \\
 & + \sum_{o=1}^O c_o^{Di} \cdot X_o^{Di}.
 \end{aligned} \tag{3}$$

To shorten notation, we omit the bounds on the parameters in the following presentation of the model constraints: For one, the capacities in each node is limited. Constraint numbers 4, 5, 6 and 7 show limitation of the capacity for suppliers, plants, distribution centers and retailers in the forward flow. Specifically, constraint number 4 is shown bound of products supplied by supplier  $i$ , constraint number 5 bounds of products produced by plant  $j$ , constraint number 6 capacity of distribution center  $k$  and constrain number 7 capacity of retailer  $l$ . Constraint number 8 and 9 shows limitation of the capacity for collection/inspection and disposal centers in the backward flow. Specifically, constraint number 8 is shown capacity of collection/inspection center  $n$  and constraint number 9 capacity of disposal center  $o$ .

$$\sum_{j=1}^J X_{ij}^{SP} \leq \bar{S}_i, \forall i \in I. \tag{4}$$

$$\sum_{k=1}^K X_{jk}^{PDc} + \sum_{l=1}^L X_{jl}^{PR} + \sum_{m=1}^M X_{jm}^{PC} \leq X_j^P \cdot \bar{P}_j, \forall j \in J. \tag{5}$$

$$\sum_{l=1}^L X_{kl}^{DcR} + \sum_{m=1}^M X_{km}^{DcC} \leq X_k^{Dc} \cdot \bar{D}c_k, \forall k \in K. \tag{6}$$

$$\sum_{m=1}^M X_{lm}^{RC} \leq X_l^R \cdot \bar{R}_l, \forall l \in L. \tag{7}$$

$$\sum_{j=1}^J X_{nj}^{CoP} + \sum_{o=1}^O X_{no}^{CoDi} \leq X_n^{Co} \cdot \bar{C}o_n, \forall n \in N. \tag{8}$$

$$\sum_{n=1}^N X_{no}^{CoDi} \leq X_o^{Di} \cdot \bar{D}i_o, \forall o \in O. \tag{9}$$

Secondly, this supply chain network is conservative so the demands of customer have to be satisfied.

$$\sum_{j=1}^J X_{jm}^{PC} + \sum_{k=1}^K X_{km}^{DcC} + \sum_{l=1}^L X_{lm}^{RC} = D_m, \forall m \in M. \tag{10}$$

Additionally, according to assumption, only a fraction  $p_m^{\text{return}}$  is returned by customers, and fraction  $p_n^{\text{disposal}}$  of the returned products has to be disposed off. Apart from these expectations, the supply chain network subjected to the law of the flow conservation in-flow and out-flow in each node must be identical. These conditions reveal:

$$\sum_{i=1}^I X_{ij}^{SP} + \sum_{n=1}^N X_{nj}^{CoP} = \sum_{k=1}^K X_{jk}^{PDc} + \sum_{l=1}^L X_{jl}^{PR} + \sum_{m=1}^M X_{jm}^{PC}, \forall j \in J. \tag{11}$$

Specifically, constraint number 11 means that the sum of the total flow of recoverable products from collection/ inspection centers and the flow of the products from suppliers to a plant is equal to the flow of the products from that plant to distribution centers, retailers, and customers in the normal delivery, direct delivery, and direct shipment.

$$\sum_{j=1}^J X_{jk}^{PDc} = \sum_{l=1}^L X_{kl}^{DcR} + \sum_{m=1}^M X_{km}^{DcC}, \forall k \in K. \tag{12}$$

$$\sum_{j=1}^J X_{jl}^{PR} + \sum_{k=1}^K X_{kl}^{DcR} = \sum_{m=1}^M X_{lm}^{RC}, \forall l \in L. \tag{13}$$

$$P_m^C \cdot \left( \sum_{j=1}^M X_{jm}^{PC} + \sum_{k=1}^K X_{km}^{DcC} + \sum_{l=1}^L X_{lm}^{RC} \right) = \sum_{n=1}^N X_{mn}^{CCo}, \forall n \in N. \tag{14}$$

$$\sum_{o=1}^O X_{no}^{CoDi} = P_n^{Co} \cdot \sum_{m=1}^M X_{mn}^{CCo}, \forall n \in N. \tag{15}$$

$$\sum_{j=1}^J X_{nj}^{CoP} = 1 - P_n^{Co} \cdot \sum_{m=1}^M X_{mn}^{CCo}, \forall n \in N. \tag{16}$$

Finally, we require the decision variables to be nonnegative or binary, respectively.

$$X_{ij}^{SP}, X_{jk}^{PDc}, X_{kl}^{DcR}, X_{lm}^{RC}, X_{mn}^{CCo}, X_{no}^{CoDi}, X_{nj}^{CoP}, X_{jl}^{PR}, X_{km}^{DcC}, X_{jm}^{PC} \in \mathbb{N} \cup \{0\}; \tag{17}$$

$$X_j^P, X_k^{Dc}, X_l^R, X_n^{Co}, X_o^{Di} \in \{0, 1\}; \tag{18}$$

$$\forall j \in J, k \in K, l \in L, n \in N, o \in O.$$

The resulting optimization problem is NP-hard. To approximate a respective solution, we apply a memetic algorithm.

### 3 Solution Approach

Generally speaking, network design problems are known as a NP hard problems [3,24,25]. Although the problem addressed in this research represents a mixed integer

linear programming (MILP), suitable solution in acceptable time cannot be carried out through standard methods. Therefore, we present a memetic algorithm, which belongs to the class of metaheuristic algorithms as the solution methodology [26]. The performance of memetic algorithm is attached to two main issues, “chromosome representation” and “Memetic operators” [27].

### 3.1 Proposed Memetic Algorithm (An Overview)

According to the above explanation, “chromosome representation” and “Memetic operators” applied in this study is presented briefly in following.

#### 3.1.1 Chromosome Representation

In this study, we are using the extended random path-based direct encoding method to generate the first population [28]. By using this method, the integrated logistics problem can be integrated into a memetic algorithms by using direct encoding. At the same time, we captured the complexity of a full graph and reduce the size of encoding and thereby computational time by developing a two segment approach. The delivery path for each customer is shown in the first segment. Also, the guide information regarding plant as well as distribution assignment is existing in the second segment. As an illustration, the representation of the extended random path-based direct encoding method in two segments is shown in Fig. 3.

The different color in this sample of chromosome representation is considered as a sample of gene unit that shows the delivery path for customer *M*. Since there are three different delivery paths, the second segment is added to clarify which way is selected. Each gene in the second segment is assigned to an integer in the range of {0, 1, 2} for plant and {0, 1} for distribution center. The detailed decoding procedure is shown in Fig. 4. The procedure of encoding by extended random path-based direct encoding is shown in Algorithm 1.

It should be noted that this encoding approach may result to infeasible solution. Therefore, if the demand of a depot exceed the capacity of a source, the depot will be connected to another source with sufficient supply which the transportation cost is minimum.

#### 3.1.2 Memetic Operators

In this work, the cost function is implemented as the fitness value, which is related to the objective function obtained by (1). This value is assigned to the decoded chro-

First segment of chromosome representation										Second segment of chromosome representation										
Delivery path to customer 1							Delivery path to customer M							Information about assigning the plants			Information about assigning the Distribution center			
<i>Co</i>	<i>Di</i>	<i>P</i>	<i>R</i>	<i>Dc</i>	<i>P</i>	<i>S</i>	<i>Co</i>	<i>Di</i>	<i>P</i>	<i>R</i>	<i>Dc</i>	<i>P</i>	<i>S</i>	<i>P<sub>i</sub></i>	...	<i>P<sub>j</sub></i>	<i>Dc<sub>1</sub></i>	...	<i>Dc<sub>k</sub></i>	
3	1	4	3	1	4	2	...	4	1	3	5	2	3	1	1	...	2	1	...	0

Fig. 3 Chromosome in the extended random path direct encoding method-based MA

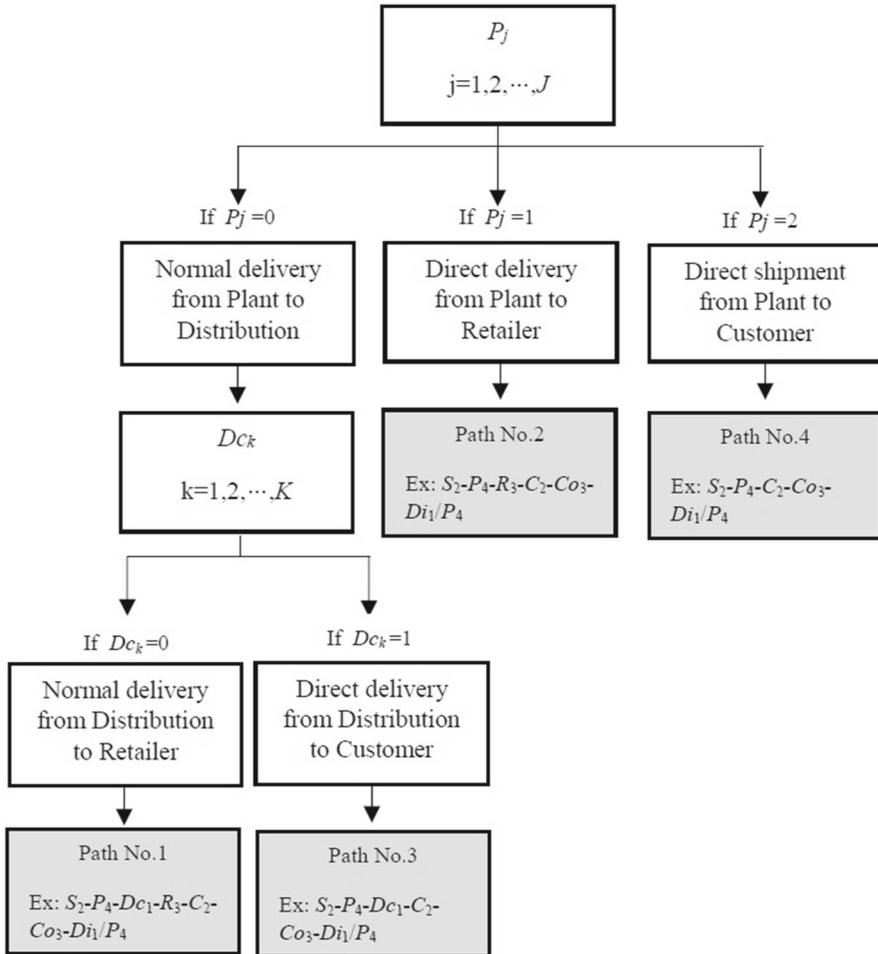


Fig. 4 Presentation of the second segment and delivery path for a sample of gene unit

mosome in order to investigate efficiency as well as accuracy of the proposed memetic algorithm.

The proposed memetic algorithm in this study has three operators including: crossover, local search and selection. For the crossover method, we select the two point crossover method which keeps the feasibility of the chromosomes. Figure 5 illustrates the two point crossover method.

Local search as one of the key factors of the memetic algorithm has an important role to explore the search space deeply. As it is impossible to swap each gene in the proposed chromosome, the applied local search technique in this paper has been updated according to the characteristics of the chromosomes. For local search technique we apply hill climbing-based random mutation method to generate more adapted offspring, reduce the chance of producing infeasible solutions and improve

**Algorithm 1** Initialization by extended random path-based direct encoding**Input:** Number of customers  $M$ Number of collection/inspection centers  $N$ Number of disposal centers  $O$ Number of plants  $J$ Number of retailers  $L$ Number of distribution centers  $K$ Number of suppliers  $I$ **Step 1**

▷ (first segment)

1 **for**  $i = 0 : M - 1$  **do**2  $ch_k[7 * i + 1] \leftarrow \text{random}(1, N)$ 3  $ch_k[7 * i + 2] \leftarrow \text{random}(1, O)$ 4  $ch_k[7 * i + 3] \leftarrow \text{random}(1, J)$ 5  $ch_k[7 * i + 4] \leftarrow \text{random}(1, L)$ 6  $ch_k[7 * i + 5] \leftarrow \text{random}(1, K)$ 7  $ch_k[7 * i + 6] \leftarrow ch_k[7 * i + 3]$ 8  $ch_k[7 * i + 7] \leftarrow \text{random}(1, I)$ 9 **end for****Step 2**

▷ (second segment, plant delivery path)

10 **for**  $i = 0 : J - 1$  **do**11  $ch_k[7 * M + i] \leftarrow \text{random}(0, 2)$ 12 **end for****Step 3**

▷ (second segment, Dc delivery path)

13 **for**  $i = 0 : K - 1$  **do**14  $ch_k[7 * M + J + i] \leftarrow \text{random}(0, 1)$ 15 **end for****Output:** Chromosome  $ch_k[\cdot]$ 

the chance of finding better solutions [28]. This adaptive local search engine is able to adjust the number of local search iterations dynamically during the process according to the size of the test problem. The following four major steps describe this method in details:

- Specify a position randomly to apply mutation for a given incumbent solution.
- Create a neighborhood by replacing the selected gene with feasible possibilities using random mutation operator [29]. The procedure of random mutation operator is simply selecting one position in the current individual and exchange the related selected gene by a regenerated randomly new number [30].
- Select the best member of neighborhood according to the fitness value as the offspring. Figure 6 illustrates an example of the local search based-mutation method, while the number of the chosen gene in the whole network is 4.
- Repeat this procedure according to the number of local search iterations. This number is considered as an input parameter of MA, and was first proposed by Hart [31]. It refers to the number of iterations of local searches during each generation cycle. For different problems, and according to the size of problems, the values of local search iterations need to be changed [32]. In this work, an adaptive local search engine is applied to adjust the number of local search iterations dynamically during the process according to the size of the considered test problems. As the number of local search iterations for the test problems should be determined by their size, we set the number of local search iterations increasingly according to

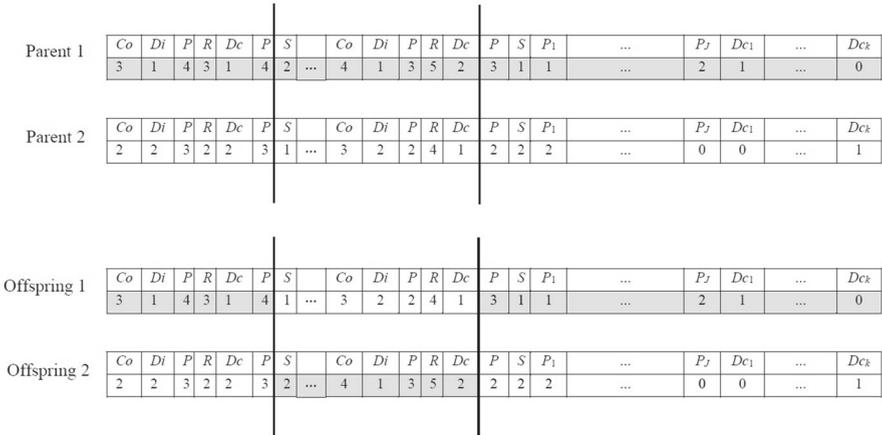


Fig. 5 Two-point crossover method

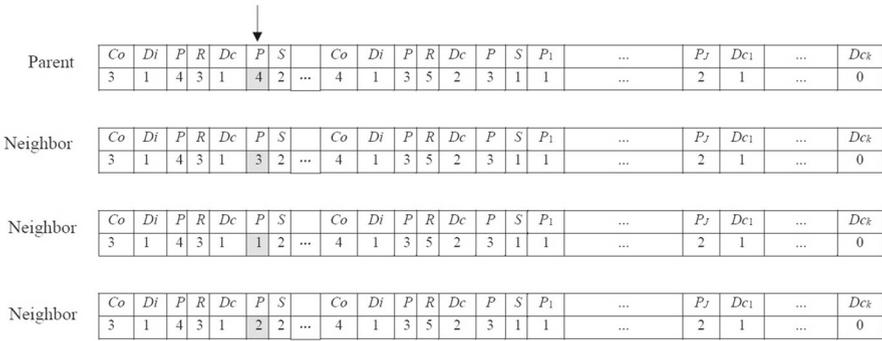


Fig. 6 Local search method

the number of retailers ( $L$ ) for all test problems. The number of retailer is selected as it has the highest number among all facilities for each test problem.

After applying local search operator, new neighborhood solutions are obtained. If the fitness value of one of the new solutions is better than the old one, the latter is replaced. An example of the applied local search is shown in Fig. 6. The local search operator is repeated according to the number of local search iteration.

For the selection part, the well-known roulette wheel method is selected [33]. The strategy applied in this method is a probabilistic selection based on fitness.

The procedure displayed a flowchart in Fig. 7 to show the steps of the solution methodology clearly. We have included different colors that shows the main focus of this study.

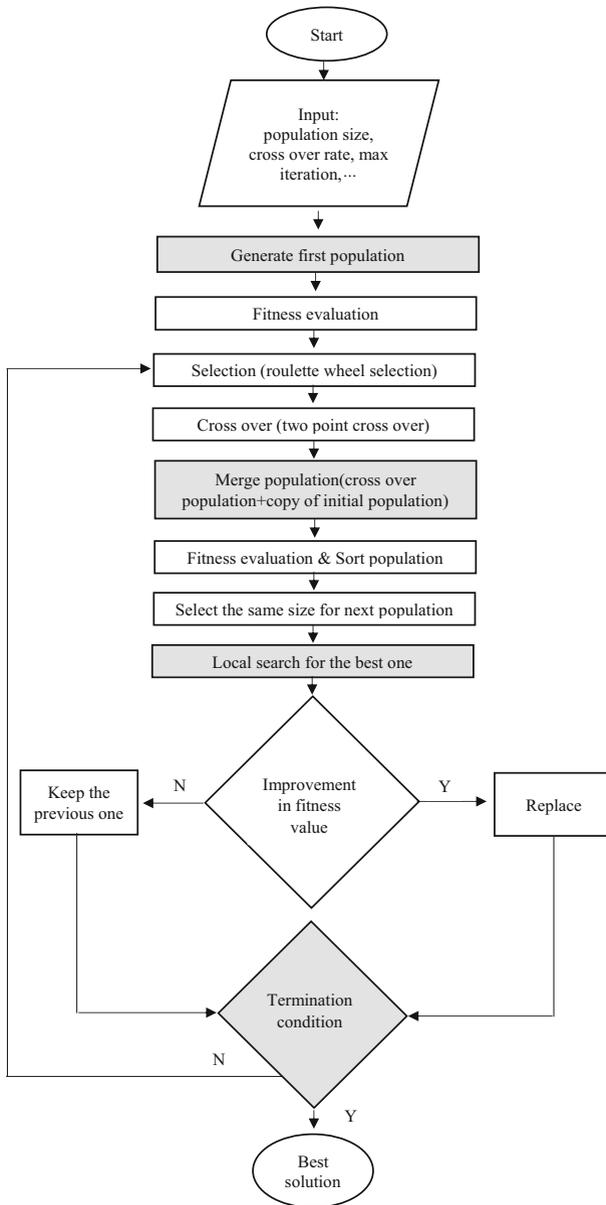


Fig. 7 Flow diagram of the proposed MA

### 4 Parameter Analysis Methods

There are many methods for studying the process parameters [34]. “One factor at a time” and “Design of Experiment (DOE)” as the most popular approaches are implemented by researchers in this field.

- **One factor at a time** “One factor at a time” also known as “monothetic analysis” is a method involving one specific condition instead of multiple factors simultaneously. In the following, the experiment are repeated by changing any other one factor.
- **Design of Experiment** “Design of Experiment (DOE)” is a statistical approach capable to determine the relationship between factors involved in a complex and multi-variable process with few trials.

Although some researchers have shown that “One factor at a time” can be more accurate some times [35], it requires more runs for the same estimation in most cases. There are two major approaches to DOE [34]:

- **Full factorial design** A “full factorial design” also known as “fully crossed design” is able to consider two or more factors. Each factor with a number of possible levels takes all possible combinations, generated by those levels across all those factors. This method allows to record the effect of each factor on the response variable individually. Using this method is time-consuming, particularly for increased number of factors. By applying this method with  $k$  factors and  $l$  levels for each factor,  $l^k$  trials are needed.
- **Taguchi method** According to the statistical theory, not every combination of parameters need to be checked [36]. In this regard, Taguchi [37] came up with a fractional replicated designs called with his name using orthogonal arrays [38]. In this method, a small set from all the possibilities is selected. By a comparison between these two methods, it is found that Taguchi’s method is more efficient than “full factorial design” approach due to same results with lesser number of experiments [34].

As Taguchi’s method found more efficient [34], we applied in this research.

## 5 Improving Optimization Using Taguchi

In order to improve operation condition of the memetic algorithm regarding the selected parameters, we applied the following steps from the Taguchi analysis [34]:

1. Identify the objective function to be optimized.
2. Identify the independence factors and number of levels for each.
3. Select a suitable orthogonal array and construct the matrix of simulation.
4. Conduct the matrix of simulation.
5. Examine and analyze data and predict the best parameter levels.
6. Confirm by simulation.

To apply the presented steps, we generated five test problems. Since the framework of the proposed logistics network in this work is not the same as in previous studies, all parameters were generated randomly using uniform distributions as given in Table 5.

Each column in Table 5 shows a number of facilities involved in each test problem separately, and also in total, optimal cost obtained by branch-and-bound algorithm from LINGO15 and number of possible routes to show how complicated they are. The number of decisions variables are 209, 234, 468, 1 006 and 1 780, respectively. Other parameters are generated randomly using uniform distributions shown in Table 6.

**Table 5** Setting of test problems

	Test no.1	Test no.2	Test no.3	Test no.4	Test no.5
Supplier	2	2	2	3	4
Plant	4	3	4	6	8
Distribution	6	8	10	15	20
Retailer	10	9	16	24	32
Customer	2	3	4	6	8
Collection/inspection	2	3	4	6	8
Disposal	1	2	2	2	4
Optimal cost	2 345	2 335	1 160	4 100	11 365
Number of facilities	27	30	42	62	84
Number of routs	178	204	456	966	1 728

**Table 6** Parameters values used in the test problems

Parameters	Range
$b_j, j \in S$	Uniform (200,1 100)
$b_j, j \in P$	Uniform (100,1 000)
$b_j, j \in Dc$	Uniform (50,900)
$b_j, j \in R$	Uniform (50,850)
$b_j, j \in D$	Uniform (100,500)
$b_j, j \in Co$	Uniform (20,100)
$b_j, j \in Di$	Uniform (20,100)
$p_j^{return}$	10%
$p_j^{disposal}$	50%
$c_{ij}$	Uniform (1,3)
$c_j, j \in P$	Uniform (100,2 500)
$c_j, j \in Dc$	Uniform (100,2 100)
$c_j, j \in R$	Uniform (100,400)
$c_j, j \in Co$	Uniform (100,500)
$c_j, j \in Di$	Uniform (50,400)

### 5.1 Identify the Objective Function to be Optimized

Taguchi suggested use of loss function for measurement. It is recommended to transform value of the loss function into a signal-to-noise ratio  $S/N = SNR$  [38]. Since the signals have a very wide dynamic range, they are mostly expressed using the logarithmic decibel.

$$\frac{S}{N} \text{ratio} = -10 \log \left[ \frac{\sum_{i=1}^n f_i^2}{2} \right], \tag{19}$$

**Table 7** Selected parameters and levels

Parameters(factors)	Level 1	Level 2	Level 3
Population size	60	100	140
Crossover rate	0.3	0.5	0.7
Local search iteration	4	8	16
Number of iteration	20	30	40

where  $f_i$  denotes the fitness value for run number  $i$  and  $n$  define the number of repeated experiments. Three different categories are classified in order to analyze the  $S/N$  ratio: “nominal is better,” “larger is better” and “smaller is better”, [39]. As the aim of this study is minimizing the total cost, “smaller is better” is considered. The best results are obtained when the total cost is minimum. This condition corresponds to the highest  $S/N$  ratio.

## 5.2 Identify the Independence Factors and Number of Level for Each

According to a comprehensive review [40–42], there are four independence factors that need to be investigated regarding the proposed algorithm. These factors are population size, crossover rate, local search iteration and number of iteration. As all of these factors have a noticeable effect on the results of the proposed model, it is important to assess all for the efficiency criterion [42]. Three different levels for each factor are considered according to the information presented in Table 7.

For population size, we considered the numbers which are the most common numbers according to the literature by respect to arithmetic progression [40]. For crossover for sure, it should be in the range of  $[0, 1]$ , and we considered 0.3, 0.5 and 0.7 to cover low rate, medium rate and high rate. For local search iteration and also number of iteration according to a comprehensive experimental research, we realized that 80 percent of progress happen in the bound we considered for the parameters [42].

## 5.3 Select a Suitable Orthogonal Array and Construct the Matrix

Taguchi method is dealing with two main tools including “ $S/N$  ratio” and a special set of arrays called “orthogonal array”. Compared to many statistical designs, orthogonal array shows a significant success. This method allows to reduce the number of experiments, while a more reliable estimation for parameter optimization can be reached. Previous studies presented many orthogonal arrays according to the problem they are facing. Still, selecting a proper orthogonal array is difficult as the reported orthogonal arrays in the literature are not able to cover all possibilities [43]. To overcome this problem, some standard orthogonal arrays have been designed according to the number of factors and levels. In order to choose an appropriate orthogonal array, the minimum number of experiment needs to be carried out first, based on the total number of freedom given as [36]:

$$N_{\text{Taguchi}} = 1 + \sum_{i=1}^{n_f} (L_i - 1) \tag{20}$$

where  $n_f$  denotes the number of factors and  $L$  define the number of levels of each factor. For the interested case, there are four different factors with three levels each, and the degree of freedom is obtained  $1 + (4 * 2) = 9$ . It means at least 9 runs need to carry out, and another selection of orthogonal array with more runs may be used to increase accuracy. Here, we utilized Minitab software to find an automatic design of the orthogonal array as well as for analysis of experiments. Minitab software suggested using  $L_9$  orthogonal array, i.e., 9 instead of  $3^4 = 81$  experiments. The details of the proper orthogonal array is shown in Table 8:

### 5.4 Conduct the Matrix Simulation

In accordance with the presented orthogonal array mentioned in Table 8, simulation were carried out with their factors and levels. The scheme with the specified values of the factors is given in Table 9. Each of these nine experiment is repeated 10 times to generate more reliable averaged results.

### 5.5 Examine and Analyze Data to Predict the Best Parameter Levels

According to the obtained results presented in [42], we realized that all the proposed parameters are important and needed to be investigated further. Since we are dealing with a strategic problem, there is no time limitation. But still having some information regarding the best values of the parameters can help us to improve our results, applicable to other types of proceeding when time limitation is considered in advance. On the other hand, as there is no similar case in the literature, here Taguchi’s method is adapted to this aim. As mentioned before, we are dealing with a strategic problem, so there is no time limitation. In the case of finding optimal parameters, we need to consider a specific time, otherwise increasing each parameters can improve

**Table 8** Orthogonal array  $L_9(3^4)$

trial	Population size	Crossover rate	Local search iteration	Number of iteration
1	1	1	1	1
2	1	2	2	2
3	1	3	3	3
4	2	1	2	3
5	2	2	3	1
6	2	3	1	2
7	3	1	3	2
8	3	2	1	3
9	3	3	2	1

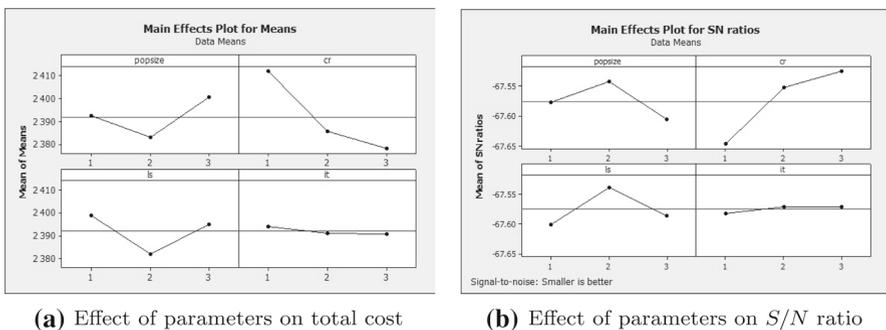
**Table 9** Orthogonal array with the specified values

trial	Population size	Crossover rate	Local search iteration	Number of iteration
1	60	0.3	4	20
2	60	0.5	8	30
3	60	0.7	16	40
4	100	0.3	8	40
5	100	0.5	16	20
6	100	0.7	4	30
7	140	0.3	16	30
8	140	0.5	4	40
9	140	0.7	8	20

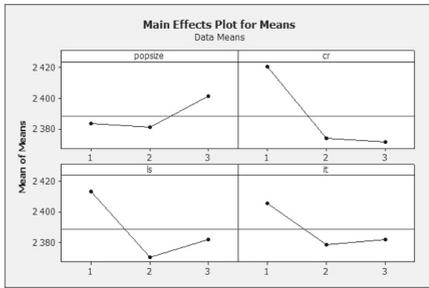
the results. In these regard, the following steps are applied. First, we need to fix an error percent which is acceptable for most of industrial players and logistics problems. This number has fixed as 0.05 ([27]). Then, we need the specific value of parameters that lead us to an error percent equal or less than 0.05 percent which are available in [42]. In the next step, a high number of iteration is assumed. This number should be more that the required number of iteration that is needed in practice. Using this way, we are able to find the average number of iteration which is required to let the algorithm find results with the preselected error percent. Afterward, the specific time that these results were obtained is recorded. Finally, the goal is discovering the best value of parameters once the algorithm is forced to find out solution in half of the recorded time using Minitab software.

Figures 8, 9, 10, 11 and 12 display the effect of the parameters on the objective function value and  $S/N$  ratio, respectively.

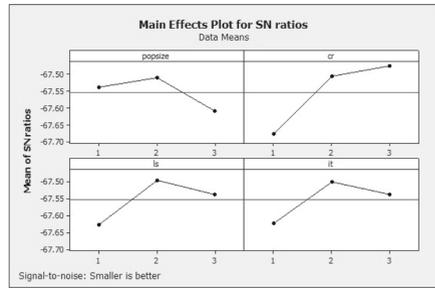
Since  $S/N$  ratio has the “large is better” characteristic, the analysis of the results leads to the population size at level 2, crossover rate at level 3, local search iteration at level 2 and number of iteration at level 2.



**Fig. 8** Problem no. 1

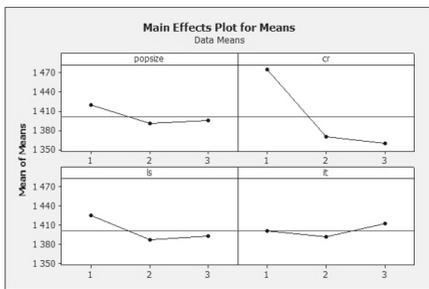


(a) Effect of parameters on total cost

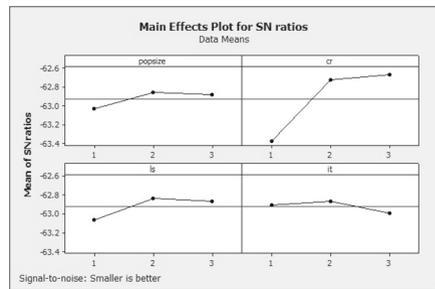


(b) Effect of parameters on  $S/N$  ratio

Fig. 9 Problem no. 2

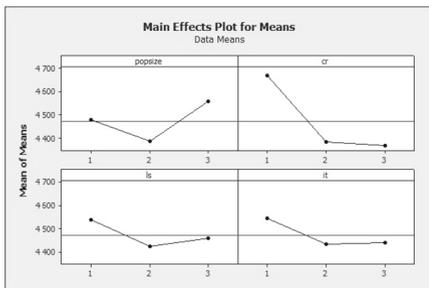


(a) Effect of parameters on total cost

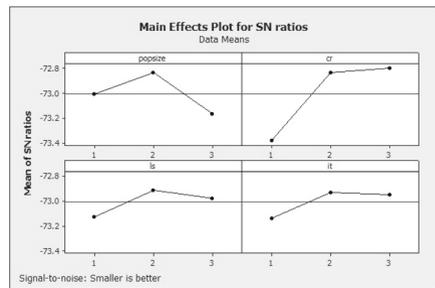


(b) Effect of parameters on  $S/N$  ratio

Fig. 10 Problem no. 3



(a) Effect of parameters on total cost



(b) Effect of parameters on  $S/N$  ratio

Fig. 11 Problem no. 4

### 5.6 Confirmation Simulation

Once the best level of the parameters were chosen according to the mentioned condition, the final step is needed to be implemented. The idea of the confirmation experiment is to validate the results obtained from the experiments. A confirmation experiment was executed 10 times with the obtained combination of parameters. The actual value of cost obtained by MATLAB was calculated based on the experiments.

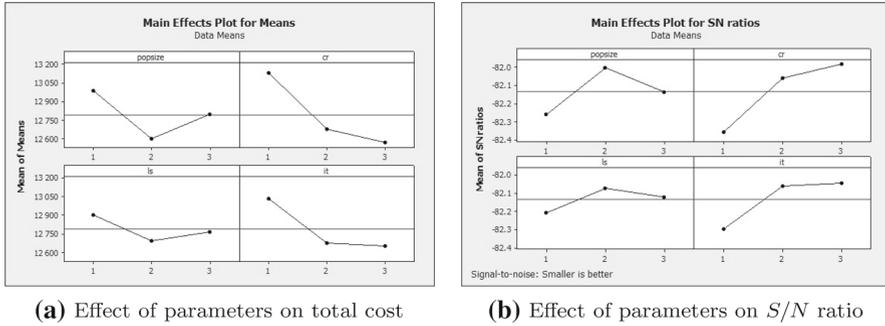


Fig. 12 Problem no. 5

The predicted value obtained by Minitab software is computed according to statistical data. Our implementation was written in MATLAB R2015b and run on the PC with Intelr CoreTM i5 2.40GHz with12 GB RAM. All results are provided in Table 10.

The results show that the predicted value of cost obtained with the best level of parameters is very close to the value of the actual cost. This validates the presented Taguchi method for predicting the optimal levels of the parameters.

### 6 Conclusions and Future Studies

This research focuses on an integrated, flexible, multi-stages, single-period and single-product supply chain network that is formulated as a mixed integer linear programming. A memetic algorithm is adapted as the solution methodology using a new chromosome representation. We applied the Taguchi method to design the optimal parameters of the proposed memetic algorithm. From the confirmation experiments, a good agreement between the predicted results and the actual results is observed. Using Taguchi method in determining the optimal level of parameters is a new contribution of this study. In the present work, only four parameters with three different levels were selected. More levels can also be considered in order to obtain better values. Different crossover, local search and selection method can be added in the analysis as well.

**Table 10** Confirmation experiment data (parameter combination Popsiz2, Cr3, Ls2, It2)

trial	Predicted value	Actual value	Error percent
1	2 346.2	2 346.8	0.000 2
2	2 336.5	2 337	0.000 2
3	1 324.67	1 325	0.000 2
4	4 188.5	4 248	0.014
5	12 168.7	12 375	0.016

**Acknowledgements** The authors would like to appreciate the support International Graduate School (IGS) of Bremen University to support, help and advice as well as the Deutscher Akademischer Austausch Dienst (DAAD) for financial support of this research under the GSSP program of the IGS.

**Funding** Open Access funding enabled and organized by Project DEAL.

## Declaration

**Conflict of interest** The authors declare that there is no conflict of interest regarding the publication of this paper.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- [1] Chan, F., Chung, S.: Multi-criteria genetic optimisation for distribution network problems. *Int. J. Adv. Manuf. Technol.* **24**, 517–532 (2004)
- [2] Selim, H., Ozkarahan, I.: A supply chain distribution network design model: an interactive fuzzy goal programming-based solution approach. *Int. J. Adv. Manuf. Technol.* **35**, 401–418 (2008)
- [3] Gen, M., Cheng, R.: *Genetic Algorithms and Engineering Design*. Wiley, New York (1997)
- [4] Dasci, A., Verter, V.: A continuous model for product-distribution system design. *Eur. J. Oper. Res.* **129**, 278–298 (2001)
- [5] Amiri, A.: Designing a distribution network in a supply chain system: formulated and efficient solution procedure. *Eur. J. Oper. Res.* **171**, 567–576 (2006)
- [6] Listes, O., Dekker, R.: A stochastic approach to a case study for product recovery network design. *Eur. J. Oper. Res.* **160**, 268–287 (2005)
- [7] Thomas, D., Griffin, P.: Coordinated supply chain management. *Eur. J. Oper. Res.* **94**, 1–15 (1996)
- [8] Pishvae, M.S., Farahani, R.Z., Dullaert, W.: A memetic algorithm for bi-objective integrated forward/reverse logistics network design. *Comput. Oper. Res.* **37**, 1100–1112 (2010)
- [9] Altıparmak, F., Gen, M., Lin, L., Paksoy, T.: A genetic algorithm approach for multi-objective optimization of supply chain networks. *Comput. Ind. Eng.* **51**, 197–216 (2006)
- [10] Zhou, G., Min, H., Gen, M.: A genetic algorithm approach to the bi-criteria allocation of customers to warehouse. *Int. J. Prod. Econ.* **86**, 35–45 (2003)
- [11] Du, F., Evans, G.: A bi-objective reverse logistics network analysis for post-sale service. *Comput. Oper. Res.* **35**, 2617–2634 (2008)
- [12] Neto, J.Q.F., Walther, G., Bloemhof, J., Nunen, A.E.E.V., Spengler, T.: From closed-loop to sustainable supply chain: the weee case. *Int. J. Prod. Res.* **48**(15), 4463–4481 (2010)
- [13] Alumur, S.A., Nickel, S., da Gamad, F.S., Verter, V.: Multi-period reverse logistics network design. *Eur. J. Oper. Res.* **220**(1), 67–78 (2012)
- [14] Fleischmann, M., Beullens, P., Bloemhof-ruwaard, J.M., Wassenhove, L.: The impact of product recovery on logistics network design. *Prod. Oper. Manag.* **10**, 156–173 (2001)
- [15] Yeh, W.: A hybrid heuristic algorithm for the multistage supply chain network problem. *Int. J. Adv. Manuf. Technol.* **26**, 675–685 (2005)
- [16] Jayaraman, V., Patterson, R., Roland, E.: The design of reverse distribution networks: model and solution procedure. *Eur. J. Oper. Res.* **150**, 128–149 (2003)
- [17] Pishvae, M.S., Kianfar, K., Karimi, B.: Reverse logistics network design simulated annealing. *Int. J. Adv. Manuf. Technol.* **47**(1), 269–281 (2010)
- [18] Chouhan, V.K., Khan, S.H., Hajiaghayi-Keshteli, M., Subramanian, S.: Multi-facility-based improved closed-loop supply chain network for handling uncertain demands. *Soft. Comput.* **24**, 7125–7147 (2020)

- [19] Gumus, D.B., Ozcan, E., Atkin, J.: An analysis of the taguchi method for tuning a memetic algorithm with reduced computational time budget, In: International Symposium on Computer and Information Sciences, (2016) [https://doi.org/10.1007/978-3-319-47217-1\\_2](https://doi.org/10.1007/978-3-319-47217-1_2)
- [20] Alavidoost, M.H., Jafarnejad, A., Babazadeh, H.: A novel fuzzy mathematical model for an integrated supply chain planning using multi-objective evolutionary algorithm. *Soft. Comput.* **25**, 1777–1801 (2021)
- [21] Akbari, M., Molla-Alizadeh-Zavardehi, S., Niroomand, S.: Meta-heuristic approaches for fixed-charge solid transportation problem in two-stage supply chain network. *Oper. Res. Int. J.* **20**, 447–471 (2020)
- [22] Zahraee, S.M., Rohani, J.M., Firouzi, A., Shahpanah, A.: Efficiency improvement of blood supply chain system using taguchi method and dynamic simulation. *Proc. Manuf.* **2**, 1–5 (2015)
- [23] Vinary, V.P., Sridharan, R.: Taguchi method for parameter design in aco algorithm for distribution-allocation in a two-stage supply chain, *Int. J. Adv. Manuf. Technol.* (2013)
- [24] Gottlieb, J., Paulmann, L.: Genetic algorithms for the fixed charge transportation problem, In: IEEE World Congress on Evolutionary Computation INN, I.C. (ed.), pp. 330–335 (1998)
- [25] Jo, J.B., Yinzheng, L., Gen, M.: Nonlinear fixed charge transportation problem by spanning tree-based genetic algorithm. *Comput. Ind. Eng.* **53**, 290–298 (2007)
- [26] Behmanesh, E., Pannek, J.: Modeling and random path-based direct encoding for a closed loop supply chain model with flexible delivery paths, In: The Seventh IFAC Conference on Management and Control of Production and Logistics, p. **49**(2), 78–83 (2016)
- [27] Wang, H., Hsu, H.: A closed-loop logistic model with a spanning-tree based genetic algorithm. *Comput. Oper. Res.* **37**, 376–389 (2010)
- [28] Behmanesh, E., Pannek, J.: A memetic algorithm with extended random path encoding for a closed-loop supply chain model with flexible delivery. *J. Logist. Res.* **9**(22), 1–12 (2016)
- [29] Michalewicz, Z.: Genetic algorithms + data Structures = evolution Programs. Springer-Verlag, Berlin (1996)
- [30] Yun, Y., Moon, C., Kim, D.: Hybrid genetic algorithm with adaptive local search scheme for solving multistage-based supply chain problems. *Comput. Ind. Eng.* **56**, 821–838 (2009)
- [31] Hart, W.: Adaptive Global Optimization with Local Search. Ph.D Thesis, University of California, San Diego, CA, (1994)
- [32] Liu, C., Li, B.: Memetic algorithm with adaptive local search depth for large scale global optimization, In: IEEE Congress on Evolutionary Computation, (2014)
- [33] Lipowski, A., Lipowska, D.: Roulette-wheel selection via stochastic acceptance. *Physica A.* **391**(6), 2193–2196 (2012)
- [34] Athreya, S., Venkatesh, Y.D.: Application of taguchi method for optimization of process parameters in improving the surface roughness of lathe facing operation. *Int. Refer. J. Eng. Sci.* **1**(3), 13–19 (2012)
- [35] Daniel, C.: One at a time plans. *J. America Atatistical Assoc.* **68**, 353–360 (1973)
- [36] Zandieh, M., Amiri, M., Vahdani, B., Soltani, R.: A robust parameter design for multi-response problems. *J. Comput. Appl. Math.* **230**, 463–476 (2009)
- [37] Taguchi, G., Konishi, S.: Taguchi Methods, Orthogonal Arrays and Linear Graphs, Tools for Quality American Supplier Institute, American Supplier Institute, <https://www.researchgate.net/publication/310478918> (1987)
- [38] Subulan, K., Tasan, A.S.: Taguchi method for analyzing the tactical planning model in a closed-loop supply chain considering remanufacturing option. *Int. J. Adv. Manuf. Technol.* **66**, 251–269 (2013)
- [39] Shula, S., Tiwari, M., Wan, H., Shankar, R.: Optimization of the supply chain network:simulation, taguchi, and psychoclonal algorithm embedded approach. *Comput. Ind. Eng.* **58**, 29–39 (2010)
- [40] Piotrowski, A.P.: Review of differential evolution population size. *Swarm Evol. Comput.* **32**, 1–24 (2017)
- [41] Lee, D., Dong, M.: A heuristic approach to logistics network design for end-of-lease computer pproduct recovery. *Transp. Res. Part E* **44**, 455–474 (2007)
- [42] Behmanesh, E., Pannek, J.: The effect of various parameters of solution methomethod on a flexible integrated supply chain model, *Mathematical Problems in Engineering.* <https://doi.org/10.1155/2018/5935268> (2018)
- [43] Hedayat, A.S., Sloane, N.J.A.: Orthogonal Arrays: Theory and Applications. Springer, New York (1999)