Int J Adv Manuf Technol (2013) 68:917–931
DOI 10.1007/s00170-013-4953-6

ORIGINAL ARTICLE

Designing and planning a multi-echelon multi-period multi-product closed-loop supply chain utilizing genetic algorithm

Hamed Soleimani · Mirmehdi Seyyed-Esfahani · Mohsen Akbarpour Shirazi

Received: 17 August 2012 / Accepted: 7 March 2013 / Published online: 14 April 2013

Abstract Designing and planning a closed-loop supply chain in a comprehensive structure is vital for its applicability. To cope with the design and planning issue of a comprehensive closed-loop supply chain network, this paper develops an extended model, which is multi-echelon, multi-product, and multi-period in a mixed integer linear programming framework. The word "comprehensive," in our mathematical approach, in designing and planning a closed-loop supply chain problem, can be analyzed from two complementary angles: including all possible entities (facilities) of a real condition and considering minimum limitations on possible flows between entities. In our proposed model, customers can be supplied via manufacturers, warehouses, and distributors, as an example. The proposed model is solved by CPLEX optimization software and by a developed genetic algorithm. During this computational analysis, we compare results of proposed pretuned genetic algorithm with a global optimum of CPLEX solver. Then, a sufficient number of large-size instances are generated and solved by the proposed genetic algorithm. To the best of our knowledge, there has been no similar multi-period multi-product closed-loop supply chain design and planning problem utilizing any kind of meta-heuristics let alone genetic algorithms. Therefore, in this issue, it is an original research, and results prove the acceptable performances of the developed genetic algorithm.

Keywords Closed-loop supply chain · Design and planning · Genetic algorithm · Mixed integer linear programming

H. Soleimani · M. Seyyed-Esfahani (✉) · M. A. Shirazi
Department of Industrial Engineering and Management Systems, Amirkabir University of Technology (Tehran Polytechnic), 424 Hafez Ave, P.O. Box: 15875-4413, Tehran, Iran
e-mail: msesfahani@aut.ac.ir

1 Introduction

A supply chain, in its classical form is a combination of processes to fulfill a customer request, and it not only includes the manufacturers and suppliers, but also contains transporters, warehouses, retailers, and customers themselves [1]. To update this definition, we need to consider environmental effects and social responsibilities, which lead us to regard reverse logistics of used products (called return products). Consequently, by trying to manage a forward and reverse supply chain simultaneously, we will have a closed-loop supply chain. In a closed-loop supply chain, manufacturers have to be responsible for collecting used products from customers and trying to reuse them in any possible form (or at least dispose them).

Recently, a combination of four main reasons actuates manufacturers toward focusing on closed-loop supply chain (CLSC) issues: governmental legislation, customer awareness, social responsibilities, and economical concerns of organizations. Roughly speaking, a closed-loop supply chain used to be an undesirable constraint in the past, but it is an acceptable necessity, and interestingly, it will be the only remedy to sustain, in the future. Consequently, if industries want to sustain, they should regard forward and reverse supply chains, together. Beamon presents a comprehensive illustration of a simple closed-loop supply chain network [2].

In dealing with a closed-loop supply chain, we have different long-term and mid-term decision steps, called design and planning, respectively. In the designing stage, strategic decisions on location of all facilities (configurations) are made. In planning, we should determine the very important part of the supply chain network, which is determining the quantity of flows between supply chain entities [1]. As the design and planning stages are interdependent, regarding integrated approaches becomes unavoidable.

The presented model in this paper deals with the two above-mentioned decision-making stages: design and planning. The main aim of this paper is to improve earlier attempts in the designing and planning of a closed-loop supply chain by presenting a comprehensive model. We will try to consider all probable entities and possible network flows in our proposed CLSC. In the literature review part of this paper, this gap is illuminated. Then, utilizing sufficient numbers of appropriate instances, the developed model will be evaluated and solved. As we deal with large-scale instances in a practical environment, a genetic algorithm (GA) is proposed to solve issue that cannot be solved by exact solvers.

The rest of this paper is arranged as follows. In Section 2, a complete literature review is presented. Model characteristics and formulations are demonstrated in Section 3. Tuning of the proposed genetic algorithm is performed in Section 4. Computational analysis is illustrated in Section 5. Finally, Section 6 presents the conclusion and future researches.

2 Literature review

We try to propose a comprehensive CLSC network, to be attained by analyzing previous trends in designing and planning a CLSC. Hence, a research in literature is aimed. The word "comprehensive," in our mathematical approach in designing and planning a CLSC problem, can be analyzed from two complementary angles:

- In the designing stage, we can expect a comprehensive model not only to encompass all real-world entities but also to release any impractical assumptions. As an example, single-period or single-product models are far from practical and real-world situations. A model, which does not include warehouses as separate entities, needs to be revised in case of implementation. We try to propose a model with minimum limitations.
- In the planning stage, we can expect a comprehensive model to regard all possible flows between entities. As an example, in practical cases, customers can be supplied not only by distributors but also by manufacturers or warehouses directly. This definitely makes some significant changes in the size of the problem and its practicability. We try to cover maximum possible network flows in the proposed model.

We attempt to propose a model that can meet the most possible real-world requirements. In order to achieve this and to prove its cogency, comprehensive research in literature is imperative and has been realized by reviewing more than 80 different papers in this field. At first, some general explanations of selected papers are presented. Then, at the end of the literature review and based on the characteristics of each paper, gaps will be summarized. In the literature survey, we try to find the models that consider acceptable real-world entities including sufficient flows.

The two fundamental papers on the closed-loop supply chain are Beamon [2] and Fleischmann [3]. Beamon worked on environmental factors toward proposing a simple structure of a CLSC. He explained the classical supply chain and closed-loop supply chain completely. His aim was to offer a conceptual framework for CLSC. Fleischmann deals with logistic network characteristics including reverse supply chain.

Schultmann [4] researched in an automotive battery industry that faces severe regulations in Germany. He used simulation in his CLSC. This paper is a case study, and it can rarely be used in other environments. Pibernik [5] coped with the problem of planning supply chain in centralized and decentralized conditions. His model considered just the planning stage of decisions, and hence, it is not an integrated approach. Focusing on designing a closed-loop supply chain, more related papers should be mentioned. French et al. [6] researched on the pertinence of CLSC in process industries. Prahinski et al. [7] tried to present a review study on logistics and its drastic effects. He confirmed the lack of an integrated comprehensive model in literature in his review. Biehl et al. [8] researched on a case study of reverse logistic in the carpet industry in the USA.

Since 2008, quantitative models for CLSC design were presented to cover gaps. Papers of Kusumastuti et al. [9] and Lee et al. [10] are two case studies of designing a CLSC in computer industries. However, they are not integrated studies. Kusumastuti researched on the designing level of CLSC and vice a versa; Lee [10] presented a model for CLSC network planning. Fuente et al. [11] presented an integrated model just for designing a CLSC in metal industries in Spain. Amaro et al. [12] also worked on a similar research like Fuente but in the pharmaceutical sector. Based on our definition, all of them lack generality.

In 2009, integrity and uncertainty were two main popular issues of CLSC studies. Different papers were published on these subjects (like Mutha et al. [13], Dehghanian and Mansour [14], Xanthopoulos [15], and Kannan [16]). Two important review papers were published in 2009: Pokharel and Mutha [17] and the paper of Subramoniam et al. [18]. The first one reviews current advancements in reverse logistics. They mention that almost all papers in this field are case based; thus, they cannot be used in other cases. Hence, it is so important to have a comprehensive model. This paper attempts to cover this gap. The second paper was a review of reverse logistics in the automotive industry.

Wang and Hsu's paper, in 2010 [19], presented a comprehensive deterministic model in which distribution centers were collection centers too (so it could not pass generality requirements as mentioned). A minimum spanning tree-based genetic algorithm was its solving approach.

El-Sayed and Afia [20] presented a single-product multi-period CLSC design. Due to its good notation, we have extended our general model based on its formulation. Surely, because of our comprehensive approach, many improvements are seen here. For instance, the proposed model of this paper is multi-products; warehouses are noticed as separate entities, and different network flows between all facilities are considered. It should be noted that this paper is not a trivial extension as it considers all possible entities and flows in a multi-period multi-product CLSC network. Consequently, the final model will be complicated and huge. Further, an appropriate genetic algorithm is developed to cope with real-sized instances.

A well-known reviewing paper of Gold et al. [21] reviewed all case studies in a sustainable supply chain management, which were published from 1994 to 2007 in English-speaking peer-reviewed journals. They mention lack of presenting and solving a comprehensive model. Subramanian et al. [22] presented a mathematical model for a multi-echelon, multi-product, and single-period closed-loop supply chain. He tried to offer a generalized single-period model. Kannan et al. [23] works on designing CLSC in a deterministic situation using genetic algorithm. Pishvaee et al. [24] utilized a robust optimization approach in dealing with a CLSC. This model was the single-product, single-period type. A noticeable review of El korchi and Millet [25] offered 18 structures for formulations. They could not conclusively segregate these structures to one or two general types. Finally, Gomes et al. [26] studied on a Portuguese company in the electronic field just in designing CLSC.

In light of the above, the necessity of proposing a comprehensive model for a closed-loop supply chain network is clarified. The entire literature review is summarized as follows:

- Between elite relative papers in literature, there are just four multi-period, multi-product papers [9, 23, 26, 27], and three of the mentioned are case-based papers; the fourth one is a graph-based model, which neither includes suppliers nor copes with large-scale instances [26]. Further, none of them considers practical-sized instances. In conclusion, the important claim about the necessity of a multi-period multi-product closed-loop supply chain model is clarified. In this paper, a multi-period multi-product closed-loop supply chain model is developed.
- To the best of our knowledge, not considering a real-sized problem is sensible. Indeed, based on solving complexities, practical-scale instances are always ignored. They are just mentioned in [4, 10, 14, 28, 29]. In order to cope with the proposed complicated model, a genetic algorithm is developed and validated for small-sized and large-scale instances.
- In reviewing literature, we find the genetic algorithms' acceptable performance in the design and planning of a closed-loop supply chain problem. Consequently, we choose this famous algorithm to deal with real-size instances [14, 16, 28, 30].

We try to present a multi-period, multi-product model, which covers almost all possible network flows between entities. For instance, customers can receive products via manufacturers, warehouses, and distributors. Such network flows are rarely seen in other researches. Roughly speaking, this paper tries to cover two main identified gaps in the CLSC design and planning problem by proposing a comprehensive multi-product multi-period model, utilizing a genetic algorithm to cope with real-sized instances.

3 Model description and formulation

In order to have a comprehensive type of model in designing and planning a closed-loop supply chain, we suggest a multi-period, multi-product, and multi-echelon model, which includes different network flows based on the practical realizations. The assumptions of the model are as follows:

- The model is multi-echelon, multi-product, and multi-period, consisting of suppliers, manufacturers, warehouses, distributors, retailers (customers), disassembly centers (collection centers), redistributors, disposal centers, and second customers (see Fig. 1).
- Potential locations, capacity of all facilities, and all cost parameters are predetermined.
- Quality, demands, and prices of return products are different from new products.
- If we cannot fulfill customers' demands; shortages cannot be covered in the next periods, so we endure shortage costs. This assumption will encourage the supply chain network to cover customers' demands. If, in reality, there is an option to cover demands in the next period, the presented model will simply cover it by changing unit shortage cost.
- Due to the generality, costs like shortage costs, holding costs, etc., are dependent on the periods and products and so are not fixed for all periods or all products.
- Manufacturers, warehouses, distributors supply first customers.
- Manufacturers, warehouses, disassembly centers and redistributors supply second customers.

The whole network flow between facilities is shown in Fig. 1. It should be mentioned that the solid lines and dashed lines, respectively, illustrate forward and reverse flows.

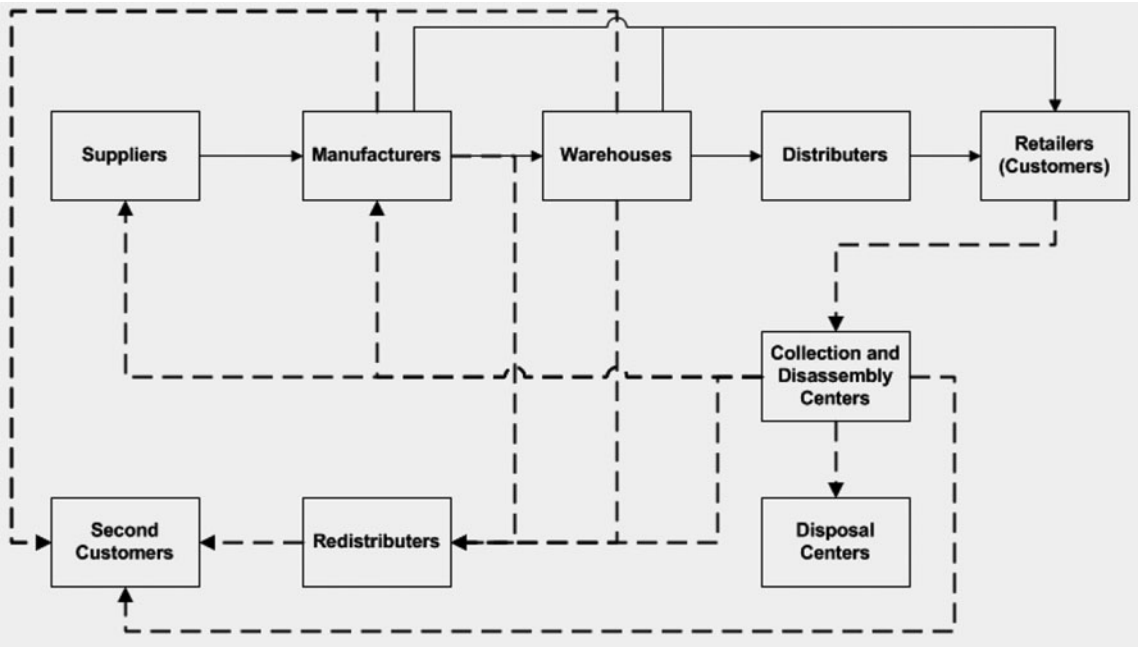

Fig. 1 The proposed model (The *arrows* show possible network flows)

Formulating a conceptual model is definitely the most important part of its applicability. The complete model is presented as follows:

Sets:

S	Potential number of suppliers, indexed by s
F	Potential number of manufacturers, indexed by f
W	Potential number of warehouses, indexed by w
D	Potential number of distributors, indexed by d
C	Potential number of the first customers (retailers), indexed by c
A	Potential number of disassembly centers, indexed by a
R	Potential number of redistributors, indexed by r
P	Potential number of disposal locations, indexed by p
K	Potential number of second customers, indexed by k
U	Number of products, indexed by u
T	Number of periods, indexed by t

Parameters:

M	A sufficiently large constant
S'	The maximum number of suppliers
F'	The maximum number of manufacturers
W'	The maximum number of warehouses
D'	The maximum number of distributors
A'	The maximum number of disassembly centers
R'	The maximum number of redistributors
P'	The maximum number of disposal centers
D_{kut}	Demand of product u of the second customer k in period t
P_{cut}	Unit price of product u at the first customer c in period t
PH_{cut}	Purchasing costs of product u at the first customer c in period t
P_{kut}	Unit price of product u at the second customer k in period t
F_i	Fixed costs of the opening location i
DS_{ij}	Distance between any two locations i and j
SC_{sut}	Capacity of supplier s of product u in period t
SRC_{sut}	Recycling capacity of supplier s of product u in period t
FC_{fut}	Manufacturing capacity in hours of facility f of product u in period t
RFC_{fut}	Remanufacturing capacity in hours of facility f of product u in period t
WC_{wut}	Capacity of warehouse w for product u in period t
DC_{dut}	Capacity of distributor d for product u in period t
AC_{aut}	Capacity of disassembly center a for product u in period t
RC_{rut}	Capacity of redistributor r for product u in period t
PC_{put}	Capacity of disposal center p for product u in period t

Mc_{sut}	Unit material costs of product u supplied by supplier s in period t
Rc_{sut}	Recycling costs of product u per unit recycled by supplier s in period t
Fc_{fut}	Manufacturing costs of product u per unit manufactured by facility f in period t
RFc_{fut}	Remanufacturing costs of product u per unit remanufactured by facility f in period t
DAc_{aut}	Disassembly costs of product u per unit disassembled by disassembly center a in period t
RPc_{aut}	Repairing costs of product u per unit repaired by disassembly center location a in period t
Pc_{aut}	Disposal costs of product u per unit disposed by disposal location p in period t
Nc_{fut}	Non-utilized manufacturing capacity costs of product u per hour of facility f in period t
RNc_{fut}	Non-utilized remanufacturing costs of product u per hour of facility f in period t
Sc_{ut}	Shortage costs of product u per unit in period t
Fh_{fu}	Manufacturing time of product u per unit in hours at facility f
RFh_{fu}	Remanufacturing time of product u per unit in hours at facility f
Rc_{sut}	Recycling costs of supplier s of product u in period t
WH_{wut}	Holding costs of product u per unit at the warehouse w in period t
DH_{dut}	Holding costs of product u per unit at distributor d stored in period t
B_{su}, B_{fu}, B_{du}, B_{au}, B_{ru}, B_{wu}, B_{cu}	Batch size of product u from supplier s, facility f, distributor d, disassembly center a, redistributor r, warehouse w, and customer c, respectively
Tc_{ut}	Transportation costs of product u per unit per kilometer in period t
RR_{ut}	Return ratio of product u at the first customers in period t
Rc	Recycling ratio
Rm	Remanufacturing ratio
Rr	Repairing ratio
Rp	Disposal ratio

Decision variables:

L_i	Binary variable equals "1" if the location i is open and "0" if otherwise.
Li_{ij}	Binary variable equals "1" if a transportation link is established between any two locations i and j.
Q_{ijut}	Flow of batches of product u from location i to location j in period t.
R_{wut}	The residual inventory of product u at warehouse w in period t.
$Rdut$	The residual inventory of product u at distributor d in period t.

3.1 Objective function

We have considered profit as the objective function, so all sales and costs are calculated:

Total sales:

Sales of all products:

First product sales:

$$\sum_{d\in D}\sum_{c\in C}\sum_{u\in U}\sum_{t\in T} Q_{dcut}B_{du}P_{cut}+\sum_{f\in F}\sum_{c\in C}\sum_{u\in U}\sum_{t\in T} Q_{fcut}B_{fu}P_{cut} + \sum_{w\in W}\sum_{c\in C}\sum_{u\in U}\sum_{t\in T} Q_{wcut}B_{wu}P_{cut} \quad (1)$$

Second product sales:

$$\sum_{r\in R}\sum_{k\in K}\sum_{u\in U}\sum_{t\in T} Q_{rkut}B_{bu}P_{kut}+\sum_{f\in F}\sum_{k\in K}\sum_{u\in U}\sum_{t\in T} Q_{fkut}B_{fu}P_{cut} + \sum_{w\in W}\sum_{k\in K}\sum_{u\in U}\sum_{t\in T} Q_{wkut}B_{wu}P_{cut} \quad (2)$$

Total costs:

Total costs = fixed costs + material costs + manufacturing costs + non-utilized capacity costs + shortage costs + purchasing costs + disassembly center costs + recycling costs + remanufacturing costs + repairing costs + disposal costs + transportation costs + inventory holding costs.

Fixed costs:

$$\sum_{s\in S} F_s L_s + \sum_{f\in F} F_f L_f + \sum_{d\in D} F_d L_d + \sum_{a\in A} F_a L_a + \sum_{r\in R} F_r L_r + \sum_{p\in P} F_p L_p + \sum_{w\in W} F_w L_w \quad (3)$$

Material costs:

$$\sum_{s\in S}\sum_{f\in F}\sum_{u\in U}\sum_{t\in T} Q_{sfut}B_{su}\mathrm{Mc}_{sut} - \sum_{a\in A}\sum_{s\in S}\sum_{u\in U}\sum_{t\in T} Q_{asut}B_{au}(\mathrm{Mc}_{sut}-\mathrm{Rc}_{sut}) \quad (4)$$

Manufacturing costs:

$$\sum_{f\in F}\sum_{d\in D}\sum_{u\in U}\sum_{t\in T} Q_{fut}B_{fu}\mathrm{Fc}_{fut}+\sum_{f\in F}\sum_{w\in W}\sum_{u\in U}\sum_{t\in T} Q_{fwut}B_{fu}\mathrm{Fc}_{fut}+ \sum_{f\in F}\sum_{c\in C}\sum_{u\in U}\sum_{t\in T} Q_{fcut}B_{fu}\mathrm{Fc}_{fut}+\sum_{f\in F}\sum_{k\in K}\sum_{u\in U}\sum_{t\in T} Q_{fkut}B_{fu}\mathrm{Fc}_{fut} \quad (5)$$

Non-utilized capacity costs (for manufacturers):

$$\sum_{f\in F}\left(\sum_{u\in U}\left(\sum_{t\in T}\left(\left(\mathrm{FC}_{fut}/\mathrm{Fh}_{fu}\right)L_f-\sum_{d\in D}\left(Q_{fdut}B_{fu}\right)-\sum_{w\in W}\left(Q_{fwut}B_{fu}\right)-\sum_{c\in C}\left(Q_{fcut}B_{fu}\right)+\sum_{w\in W}\sum_{r\in R}Q_{wrut}B_{wu}+\sum_{w\in W}\sum_{k\in K}Q_{wkut}B_{wu}\right)\mathrm{Nc}_{fut}\right)\right)$$
$$+\sum_{f\in F}\left(\sum_{u\in U}\left(\sum_{t\in T}\left(\left(\mathrm{RFC}_{fut}/\mathrm{RFh}_{fu}\right)L_f-\sum_{r\in R}\left(Q_{frut}B_{fu}\right)-\sum_{k\in K}\left(Q_{fkut}B_{fu}\right)-\sum_{w\in W}\sum_{r\in R}Q_{wrut}B_{wu}+\sum_{w\in W}\sum_{k\in K}Q_{wkut}B_{wu}\right)\mathrm{RNc}_{fut}\right)\right) \tag{6}$$

Shortage costs (for distributor):

$$\left(\sum_{c\in C}\left(\sum_{u\in U}\left(\sum_{t\in T}\left(\sum_{t-1}^{t}D_{cut}-\sum_{t-1}^{t}\sum_{d\in D}Q_{dcut}B_{du}-\sum_{t-1}^{t}\sum_{f\in F}Q_{fcut}B_{fu}-\sum_{t-1}^{t}\sum_{w\in W}Q_{wcut}B_{wu}\right)\mathrm{Sc}_{ut}\right)\right)\right) \tag{7}$$

Purchasing costs:

$$\sum_{c\in C}\sum_{a\in A}\sum_{u\in U}\sum_{t\in T}Q_{caut}\mathrm{PH}_{cut}\mathrm{B}_{cu} \tag{8}$$

Disassembly center costs:

$$\sum_{c\in C}\sum_{a\in A}\sum_{u\in U}\sum_{t\in T}Q_{caut}B_{cu}\mathrm{DAc}_{aut} \tag{9}$$

Recycling costs:

$$\sum_{a\in A}\sum_{s\in S}\sum_{u\in U}\sum_{t\in T}Q_{asut}B_{au}\mathrm{Rc}_{sut} \tag{10}$$

Remanufacturing costs:

$$\sum_{a\in A}\sum_{f\in F}\sum_{u\in U}\sum_{t\in T}Q_{afut}B_{au}\mathrm{RFc}_{fut} \tag{11}$$

Repairing costs:

$$\sum_{a\in A}\sum_{r\in R}\sum_{u\in U}\sum_{t\in T}Q_{arut}B_{au}\mathrm{RPc}_{\mathrm{aut}} \tag{12}$$

Disposal costs:

$$\sum_{a\in A}\sum_{p\in P}\sum_{u\in U}\sum_{t\in T}Q_{aput}B_{au}\mathrm{Pc}_{put} \tag{13}$$

Transportation costs:

$$\sum_{t\in T}\sum_{u\in U}\sum_{s\in S}\sum_{f\in F}Q_{sfut}B_{su}\mathrm{Tc}_{ut}\mathrm{DS}_{sf}+\sum_{t\in T}\sum_{u\in U}\sum_{f\in F}\sum_{d\in D}Q_{fdut}B_{fu}\mathrm{Tc}_{ut}\mathrm{DS}_{fd}+\sum_{t\in T}\sum_{u\in U}\sum_{f\in F}\sum_{w\in W}Q_{fwut}B_{fu}\mathrm{Tc}_{ut}\mathrm{DS}_{fw}$$
$$\sum_{t\in T}\sum_{u\in U}\sum_{f\in F}\sum_{c\in C}Q_{fcut}B_{fu}\mathrm{Tc}_{ut}\mathrm{DS}_{fc}+\sum_{t\in T}\sum_{u\in U}\sum_{f\in F}\sum_{k\in K}Q_{fkut}B_{fu}\mathrm{Tc}_{\mathrm{ut}}\mathrm{DS}_{fk}+\sum_{t\in T}\sum_{u\in U}\sum_{w\in W}\sum_{c\in C}Q_{wcut}B_{wu}\mathrm{Tc}_{ut}\mathrm{DS}_{wc}$$
$$\sum_{t\in T}\sum_{u\in U}\sum_{w\in W}\sum_{k\in K}Q_{wkut}B_{wu}\mathrm{Tc}_{ut}\mathrm{DS}_{wk}+\sum_{t\in T}\sum_{u\in U}\sum_{d\in D}\sum_{c\in C}Q_{dcut}B_{du}\mathrm{Tc}_{ut}\mathrm{DS}_{dc}+\sum_{t\in T}\sum_{u\in U}\sum_{a\in A}\sum_{s\in S}Q_{asut}B_{au}\mathrm{Tc}_{ut}\mathrm{DS}_{as}$$
$$\sum_{t\in T}\sum_{a\in A}\sum_{u\in U}\sum_{f\in F}Q_{afut}B_{au}\mathrm{Tc}_{ut}\mathrm{DS}_{af}+\sum_{t\in T}\sum_{u\in U}\sum_{a\in A}\sum_{p\in P}Q_{aput}B_{au}\mathrm{Tc}_{ut}\mathrm{DS}_{ap}+\sum_{t\in T}\sum_{u\in U}\sum_{a\in A}\sum_{r\in R}Q_{arut}B_{au}\mathrm{Tc}_{ut}\mathrm{DS}_{ar}$$
$$\sum_{t\in T}\sum_{u\in U}\sum_{f\in F}\sum_{r\in R}Q_{frut}B_{fu}\mathrm{Tc}_{ut}\mathrm{DS}_{fr}+\sum_{t\in T}\sum_{u\in U}\sum_{w\in W}\sum_{r\in R}Q_{wrut}B_{wu}\mathrm{Tc}_{ut}\mathrm{DS}_{wr}+\sum_{t\in T}\sum_{u\in U}\sum_{r\in R}\sum_{k\in K}Q_{rkut}B_{ru}\mathrm{Tc}_{ut}\mathrm{DS}_{ruk}$$
$$\sum_{t\in T}\sum_{u\in U}\sum_{c\in C}\sum_{a\in A}Q_{caut}B_{cu}\mathrm{Tc}_{ut}\mathrm{DS}_{ca}+\sum_{t\in T}\sum_{u\in U}\sum_{w\in W}\sum_{d\in D}Q_{wdut}B_{wu}\mathrm{Tc}_{ut}\mathrm{DS}_{wd} \tag{14}$$

Inventory holding costs:

$$\sum_{w\in W}\sum_{u\in U}\sum_{t\in T}R_{wut}\mathrm{WH}_{wut}+\sum_{d\in D}\sum_{u\in U}\sum_{t\in T}R_{dut}\mathrm{DH}_{dut} \tag{15}$$

3.2 Constraints

All constraints of the proposed comprehensive model are represented as follows. Clearly, this general model is a mixed integer linear programming one:

$$\sum_{s\in S}Q_{sfut}B_{su}=\sum_{d\in D}Q_{fdut}B_{fu}+\sum_{w\in W}Q_{fwut}B_{fu}+\sum_{c\in C}Q_{fcut}B_{fu},\forall t\in T,\forall u\in U,\forall f\in F \tag{16}$$

$$\sum_{f\in F}Q_{fwut}B_{fu}+R_{wu(t-1)}=R_{wut}+\sum_{d\in D}Q_{wdut}B_{wu}+\sum_{c\in C}Q_{wcut}B_{wu}+\sum_{k\in K}Q_{wkut}B_{wu},\forall t\in T,\forall u\in U,\forall w\in W \tag{17}$$

$$\sum_{f\in F} Q_{fdut}B_{fu} + \sum_{w\in W} Q_{wdut}B_{wu} + R_{du(t-1)} = R_{dut} + \sum_{c\in C} Q_{dcut}B_{du}, \forall t \in T, \forall u \in U, \forall d \in D \quad (18)$$

$$\sum_{d\in D} Q_{dcut}B_{du} + \sum_{f\in F} Q_{fcut}B_{fu} + \sum_{w\in W} Q_{wcut}B_{wu} \le D_{cut} + \sum_{1}^{t} D_{cu(t-1)} - \left(\sum_{d\in D}\sum_{1}^{t} Q_{dcu(t-1)}B_{du} + \sum_{f\in F}\sum_{1}^{t} Q_{fcu(t-1)}B_{fu} + \sum_{w\in W}\sum_{1}^{t} Q_{wcu(t-1)}B_{wu}\right), \forall t \in T, \forall u \in U, \forall c \in C \quad (19)$$

$$\sum_{a\in A} Q_{caut}B_{cu} \le \left(\sum_{d\in D} Q_{dcut}B_{du} + \sum_{f\in F} Q_{fcut}B_{fu} + \sum_{w\in W} Q_{wcut}B_{wu}\right) RR_{ut}, \forall t \in T, \forall u \in U, \forall c \in C \quad (20)$$

$$\sum_{c\in C} Q_{caut}B_{cu} = \sum_{s\in S}(Q_{asut}B_{au}) + \sum_{f\in F}(Q_{afut}B_{au}) + \sum_{r\in R}(Q_{arut}B_{au}) + \sum_{p\in P}(Q_{aput}B_{au}), \forall t \in T, \forall u \in U, \forall a \in A \quad (21)$$

$$\sum_{c\in C}(Q_{caut}B_{cu})\mathrm{Rc} = \sum_{s\in S}(Q_{asut}B_{au}), \forall t \in T, \forall u \in U, \forall a \in A \quad (22)$$

$$\sum_{c\in C}(Q_{caut}B_{cu})\mathrm{Rm} = \sum_{f\in F}(Q_{afut}B_{au}), \forall t \in T, \forall u \in U, \forall a \in A \quad (23)$$

$$\sum_{c\in C}(Q_{caut}B_{cu})\mathrm{Rr} = \sum_{r\in R}(Q_{arut}B_{au}), \forall t \in T, \forall u \in U, \forall a \in A \quad (24)$$

$$\sum_{c\in C}(Q_{caut}B_{cu})\mathrm{Rp} = \sum_{p\in P}(Q_{aput}B_{au}), \forall t \in T, \forall u \in U, \forall a \in A \quad (25)$$

$$\sum_{a\in A}(Q_{afut}B_{au}) = \sum_{r\in R}(Q_{frut}B_{fu}) + \sum_{k\in K}(Q_{fkut}B_{fu}) + \sum_{w\in w}\sum_{k\in K}(Q_{wkut}B_{wu}) + \sum_{w\in w}\sum_{r\in R}(Q_{wrut}B_{wu}), \forall t \in T, \forall u \in U, \forall f \in F \quad (26)$$

$$\sum_{a\in A}(Q_{arut}B_{au}) + \sum_{f\in F}(Q_{frut}B_{fu}) + \sum_{w\in W}(Q_{wrut}B_{wu}) = \sum_{k\in K}(Q_{rkut}B_{ru}), \forall t \in T, \forall u \in U, \forall r \in R \quad (27)$$

$$\sum_{r\in R}(Q_{rkut}B_{ru}) \le D_{kut}, \forall t \in T, \forall u \in U, \forall k \in K \quad (28)$$

$$\mathrm{Rc} + \mathrm{Rm} + \mathrm{Rr} + \mathrm{Rp} = 1 \quad (29)$$

Constraints (16) to (28) are balanced constraints. Reviewing Fig. 1 reveals the necessity of balancing each entity. Indeed, at each node, all products entering flows per period should be equal to all issuing flows of that node for the same product in the same period. Certainly, for all the entities in Fig. 1, constraints should be set. Therefore, constraints (16) are balance constraints of manufacturers, constraints (17) to (21) are for warehouses (17), distributors (18), customers (19), disassembly centers' inputs (20), and disassembly centers output (21), respectively. Again, constraints (22) to (28) are recycling rate constraints (22), remanufacturing rate constraints (23), repairing rate constraints (24), disposal rate constraints (25), manufacturers reverse flows (26), redistributors (27), and ultimately, second customers balance constraints (28). The sum of all assigning rates via disassembly centers should be equal to one (constraint 29).

$$\sum_{f\in F} Q_{sfut}B_{su} \le SC_{sut}L_s, \forall t \in T, \forall u \in U, \forall s \in S \quad (30)$$

$$\left(\sum_{d\in D} Q_{fdut}B_{fu} + \sum_{w\in W} Q_{fwut}B_{fu} + \sum_{c\in C} Q_{fcut}B_{fu} + \sum_{k\in K} Q_{fkut}B_{fu}\right) \mathrm{Fh}_{fu} \le \mathrm{FC}_{fut}L_f, \forall t \in T, \forall u \in U, \forall f \in F \quad (31)$$

$$R_{wut} \le \mathrm{SC}_{wut}L_w, \forall t \in T, \forall u \in U, \forall w \in W \quad (32)$$

$$\sum_{f\in F} Q_{fdut}B_{fu} + \sum_{w\in W} Q_{wdut}B_{wu} + R_{du(t-1)} \le \mathrm{DC}_{dut}L_d, \forall t \in T, \forall u \in U, \forall d \in D \quad (33)$$

$$\sum_{s\in S} Q_{asut}B_{au} + \sum_{f\in F} Q_{afut}B_{au} + \sum_{r\in R} Q_{arut}B_{au} + \sum_{p\in P} Q_{aput}B_{au} \le \mathrm{AC}_{aut} \times L_a, \forall t \in T, \forall u \in U, \forall a \in A \quad (34)$$

$$\sum_{k\in K} Q_{rkut}B_{ru} \le \mathrm{RC}_{rut} \times L_r, \forall t \in T, \forall u \in U, \forall r \in R \quad (35)$$

$$\sum_{a\in A} Q_{asut}B_{au} \le \mathrm{SRC}_{sut} \times L_s, \forall t \in T, \forall u \in U, \forall s \in S \quad (36)$$

$$\sum_{a\in A} Q_{aput}B_{au} \le \mathrm{PC}_{put} \times L_p, \forall t \in T, \forall u \in U, \forall p \in P \quad (37)$$

$$\sum_{f\in F} Q_{fwut}B_{fu} \le \mathrm{WC}_{wut} \times L_w, \forall t \in T, \forall u \in U, \forall w \in W \quad (38)$$

Constraints (30) to (38) are capacity constraints, which control the maximum flows that can enter/issue from each node.

Constraint (30) controls all suppliers' output capacity for each product in all periods. Constraints (31) to (38) are for capacity of manufacturers, warehouses, distributors, redistributors, suppliers, disposal centers, and warehouses inputs.

$$\mathrm{Li}_{sf} \le \sum_{u\in U}\sum_{t\in T} Q_{sfut} \le M\,\mathrm{Li}_{sf}, \forall s \in S, \forall f \in F \quad (39)$$

$$\mathrm{Li}_{fd} \le \sum_{u\in U}\sum_{t\in T} Q_{fdut} \le M\,\mathrm{Li}_{fd}, \forall f \in F, \forall d \in D \quad (40)$$

$$\mathrm{Li}_{fw} \le \sum_{u\in U}\sum_{t\in T} Q_{fwut} \le M\,\mathrm{Li}_{fw}, \forall f \in F, \forall w \in W \quad (41)$$

$$\mathrm{Li}_{fc} \le \sum_{u\in U}\sum_{t\in T} Q_{fcut} \le M\,\mathrm{Li}_{fc}, \forall f \in F, \forall c \in C \quad (42)$$

$$\mathrm{Li}_{fk} \le \sum_{u\in U}\sum_{t\in T} Q_{fkut} \le M\,\mathrm{Li}_{fk}, \forall f \in F, \forall k \in K \quad (43)$$

$$\mathrm{Li}_{fr} \le \sum_{u\in U}\sum_{t\in T} Q_{frut} \le M\,\mathrm{Li}_{fr}, \forall r \in R, \forall f \in F \quad (44)$$

$$\mathrm{Li}_{wd} \le \sum_{u\in U}\sum_{t\in T} Q_{wdut} \le M\,\mathrm{Li}_{wd}, \forall w \in W, \forall d \in D \quad (45)$$

$$\mathrm{Li}_{wc} \le \sum_{u\in U}\sum_{t\in T} Q_{wcut} \le M\,\mathrm{Li}_{wc}, \forall w \in W, \forall c \in C \quad (46)$$

$$\mathrm{Li}_{wk} \le \sum_{u\in U}\sum_{t\in T} Q_{wkut} \le M\,\mathrm{Li}_{wk}, \forall w \in W, \forall k \in K \quad (47)$$

$$\mathrm{Li}_{wr} \le \sum_{u\in U}\sum_{t\in T} Q_{wrut} \le M\,\mathrm{Li}_{wr}, \forall w \in W, \forall r \in R \quad (48)$$

$$\mathrm{Li}_{dc} \le \sum_{u\in U}\sum_{t\in T} Q_{dcut} \le M\,\mathrm{Li}_{dc}, \forall d \in D, \forall c \in C \quad (49)$$

$$\mathrm{Li}_{ca} \le \sum_{u\in U}\sum_{t\in T} Q_{caut} \le M\,\mathrm{Li}_{ca}, \forall a \in A, \forall c \in C \quad (50)$$

$$\mathrm{Li}_{as} \le \sum_{u\in U}\sum_{t\in T} Q_{asut} \le M\,\mathrm{Li}_{as}, \forall s \in S, \forall a \in A \quad (51)$$

$$\mathrm{Li}_{af} \le \sum_{u\in U}\sum_{t\in T} Q_{afut} \le M\,\mathrm{Li}_{af}, \forall f \in F, \forall a \in A \quad (52)$$

$$\mathrm{Li}_{ar} \le \sum_{u\in U}\sum_{t\in T} Q_{arut} \le M\,\mathrm{Li}_{ar}, \forall r \in R, \forall a \in A \quad (53)$$

$$\mathrm{Li}_{ap} \le \sum_{u\in U}\sum_{t\in T} Q_{aput} \le M\,\mathrm{Li}_{ap}, \forall p \in P, \forall a \in A \quad (54)$$

$$\mathrm{Li}_{rk} \le \sum_{u\in U}\sum_{t\in T} Q_{rkut} \le M\,\mathrm{Li}_{rk}, \forall k \in K, \forall r \in R \quad (55)$$

Constraints (39) to (55) manage links between all nodes. As the left sides of constraints (39) are considered, if there are no flows between a supplier and a manufacturer of all products in all periods, then, there should be no link between these two entities. Again, based on the right side of the same constraint, if there is no real link or shipping between these two suppliers and manufacturers, we definitely cannot have any network flows here. These constraints guarantee there are no links between nodes without any actual real flows and no flows between two nodes without any actual link.

$$\sum_{s\in S} L_s \le S' \quad (56)$$

$$\sum_{f\in F} L_f \le F' \quad (57)$$

$$\sum_{d\in D} L_d \le D' \quad (58)$$

$$\sum_{w\in W} L_w \le W' \quad (59)$$

$$\sum_{a\in A} L_a \le A' \quad (60)$$

$$\sum_{r\in R} L_r \le R' \quad (61)$$

$$\sum_{p\in P} L_p \le P' \quad (62)$$

Constraints (56) to (62) manage the maximum number of allowable locations. There should be some limitations on the number of activated locations. As a result, these constraints cope with the mentioned limitation and do not let the supply chain network establish extra nodes.

In order to solve the developed NP-hard problem, a genetic algorithm is proposed. In the next two sections, the characteristics of developed GA are explained.

4 Solution methodology

As proved, traditional techniques are not efficient when search space is too large (like CLSC design and planning problem).

Table 1 Parameters of computational study

Row	Parameter	Uniform distribution or rate
1.	Demands	0–3,000
2.	Second demands rate	50 % of demand
3.	Prices	15,000–20,000
4.	Second product prices	50 % of price
5.	Purchasing costs	10 % of price
6.	Manufacturer capacity	6,000–14,000
7.	Remanufacturer capacities	50 % of manufacturer capacity
8.	Supplier capacities	18,000–42,000
9.	Supplier recycling capacities	50 % of supplier capacity
10.	Recycling costs	10–100
11.	All other reverse costs	10–100
12	Other facilities' capacities	6,000–14,000
13	Material costs	100–1,000
14	Manufacturing costs	100–1,000
15.	All other forward costs	100–1,000
16	Shortage costs	1,000–5,000
17.	Supplier fixed costs	7–10 million
18.	Manufacturer fixed costs	70–150 million
19.	Distributor fixed costs	1–2 million
20.	Warehouse fixed costs	0.1–1 million
21.	Disassembly center fixed costs	0.1–1 million
22.	Redistributors fixed costs	0.1–1 million
23.	Disposal centers fixed costs	0.1–1 million
24.	Batch size	1

For instance, if we consider just five units of each entity in the developed model, we will have 14,482 constraints and 11,336 decision variables (including 460 binary variables). Based on the vast studies, a genetic algorithm is proposed to cope with large-scale instances. In this section, the GA-based solution methodology is described. GA, like other meta-heuristics, does not have any special criteria to set parameters. Therefore, we try to set the parameters of the proposed genetic algorithm. In order to have an efficient genetic algorithm, parameters like mutation rate, population size, and iteration numbers should be set correctly. Consequently, the parameter-setting procedure is also presented in this section.

4.1 Proposed genetic algorithm

Genetic algorithms are popular meta-heuristic optimization techniques originally developed by Holland [31], based on the genetic procedure of the human body and "elite of the fittest" in Darwin's theory. Subsequently, it was applied to optimization problems in biology, engineering, and operations research. In the 1990s, different aspects of the GA were expanded; Hartl [32] researched on convergence of this algorithm, and Radcliffe [33] and Bean [34] studied crossover operator. Miller and Goldberg [35] researched on the selection strategies, and finally, Vose [36] and Koza [37], developing genetic programming, investigated the concept of basic GA of Holland. More information about GAs and other evolutionary algorithms are found in Reeves [38], Mitchell [39], and Baeck's [40] researches.

Some researches deal with closed-loop supply chain design and planning problem utilizing genetic algorithm. These are presented in the literature survey section. Whenever researchers try to cope with large-scale problems, meta-heuristics, especially genetic algorithms, are the best choices.

Table 2 characteristics of parameter-setting instance

Number of								
Suppliers	Warehouses	Distributors, redistributors	Manufacturers	Disposal, disassembly centers	Customers	Second customers	Periods	Products
25	25	25	8	8	25	25	12	4

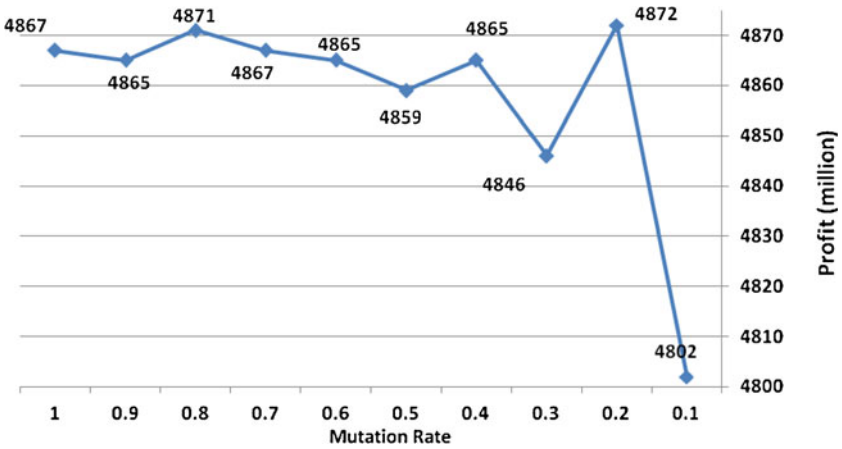

Fig. 2 Setting the "mutation rate" with population size 100 in 1,000 iterations

[14] and [28] are two of those mentioned, although there are no similar multi-period multi-product CLSC design and planning problem utilizing any meta-heuristics (let alone genetic algorithms). Therefore, this research is original in this field.

Genetic algorithm starts with a random population of solutions (called chromosomes) and attempts to improve solutions through a number of iterations (called generations). The performance of each solution is evaluated by a fitness function that corresponds to the objective function of the optimization problem. Following parental selection, crossover and mutation operators are applied. Crossover combines materials from parents to produce their children. On the other hand, mutation makes small local changes of feasible solutions to provide diversity in the population for a wider exploration of feasible solutions. The mutation operator is usually defined to ensure that the generated solutions will not be trapped in some local minimum. Moreover, because the final solution is independent to initial solutions, in most cases, the basic population is randomly generated.

Total steps of the proposed algorithm are presented as:

1. Make an initial random population (chromosomes) as basic starting solutions (for example, 100 initial random solutions).
2. Calculate fitness for all solutions according to the objective function of the model.
3. Using RAR operator [27] for crossover process, generate children via the whole current population (all parents). This operator produces one child from each couple of parents (make 50 children in the presented example).
4. Do mutation operator by a limited predetermined random rate (set at "0.2").
5. Calculate fitness for all solutions.
6. Choose best solutions based on the population size (100 in the present example) between all new children and parents to shape the next generation (the best first 100 solutions will form the next generation).
7. Repeat from line 3.

About the crossover operator (RAR), it should be mentioned that this is a well-known strategy in crossover. The rules are simple. If both of parents contain a special activated location, then that location will be activated in their produced child and vice versa. If one parent has a special entity, which the other lacks, then the final value is randomly assigned between these two.

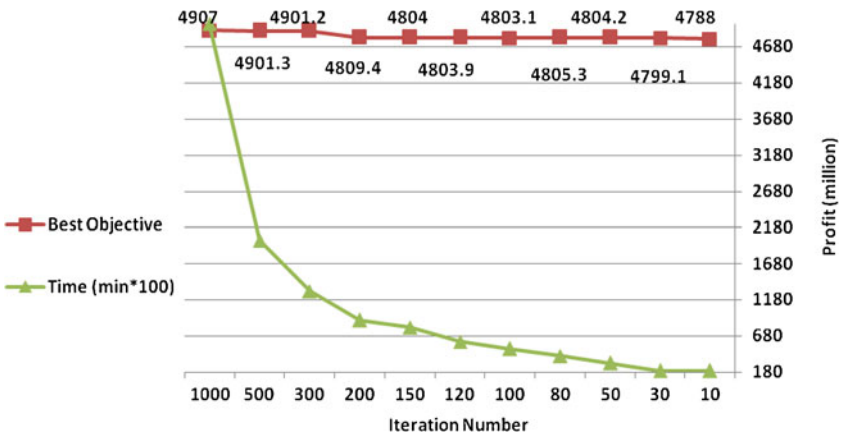

Fig. 3 Setting the "iteration number" with mutation rate 0.2 and population size 100

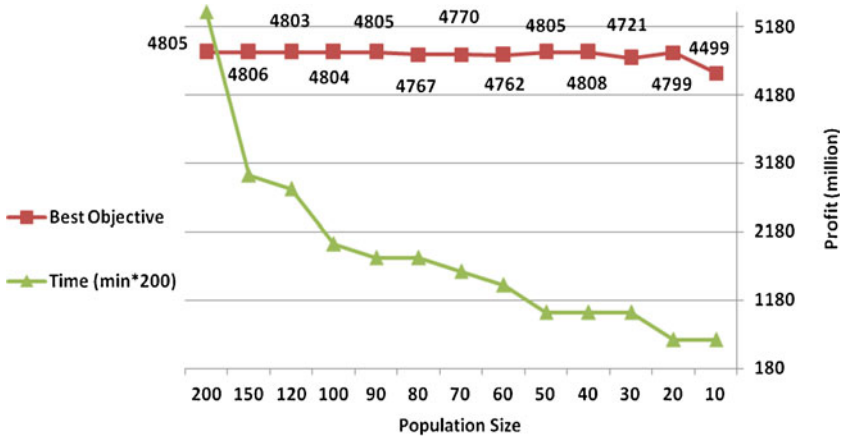

Fig. 4 Setting the "population size" with mutation rate 0.2 in 1,000 iterations

4.2 Parameter setting of the proposed genetic algorithm

Genetic algorithms have some parameters like iteration number, population size, crossover rate, and mutation rate, which can lead to different solutions when changed. Consequently, these parameters should set with some proper instances. First, we generate a mid-size instance, and then we run it many times and finally analyze GA performance in different parameter settings.

The genetic algorithm is coded by Matlab 7.12 (R2011a) software. All computations are run by a Core 2 Duo-2.26 GHz processor laptop. In all computational analyses, parameters are generated by uniformly distributed functions, which are presented in Table 1. Characteristics of selected instances of this section are illustrated in Table 2. It should be noted that recycling, remanufacturing, repair, and disposal rates are 0.2, 0.4, 0.3, and 0.1, respectively. Batch size values will be set as one.

Figures 2, 3, and 4 illustrate analysis of the first evaluations. First, we set population size and iteration number at possible level values, which are 100 and 1,000, respectively. Then, we run the instance of Table 2 many times and record the average of objective functions. Finally, we find the best mutation rate value. Then, with the achieved mutation rate, we find the best values for iteration number and population size parameters. These results are achieved by the average of several runs in each instance.

Reviewing Figs. 2, 3, and 4 helps us to set the parameters of the proposed genetic algorithm. As illustrated, "0.2" is the best value for mutation rate. About iteration number, we do not find any significant changes after "100" iterations. Hence, regarding the time criterion, we set the iteration number at "120," which guarantees achieving the best possible solution in a feasible time. In order to judge population size, we will consider two candidate values:"20" and "40." These are the most reasonable values regarding objective function and time.

We have a complete validation process based on our model. The procedure has two steps: small-sized and large-sized analyses explained in the next section.

5 Computational analysis

In order to evaluate our genetic algorithm, two validating processes are arranged: considering small- and large-scale instances. First, comparing with global optimum, a small-sized instance is developed. Afterward, the performance of the proposed genetic algorithm is compared with global optimum points, achieved by IBM ILOG CPLEX 12.2 optimization software. This process can validate the proposed GA. Then, the proposed genetic algorithm solves different kinds of large-sized instances. Details are presented in the following section.

5.1 Small-sized instance analysis

In order to evaluate effectiveness of the proposed algorithm, we first generate small but acceptable-size instance containing five units of each entity. Hence, we have five suppliers, five manufacturers, five warehouses, etc., for five products in five periods. In this phase of validation, the generated instance will run 11 times, and the objective function of CPLEX (global

Table 3 Results of small-sized problems (in millions)

Run	CPLEX global optimum	Genetic algorithm	Differences to optimum
01	3,309	2,989	10 %
02	3,400	3,082	9 %
03	3,263	2,801	14 %
04	3,274	2,740	16 %
05	3,126	2,759	12 %
06	3,450	3,109	10 %
07	3,276	2,845	13 %
08	3,304	2,754	17 %
09	3,165	2,866	9 %
10	2,191	1,971	10 %
11	5,453	5,043	8 %
Average	3,382	2,996	11 %

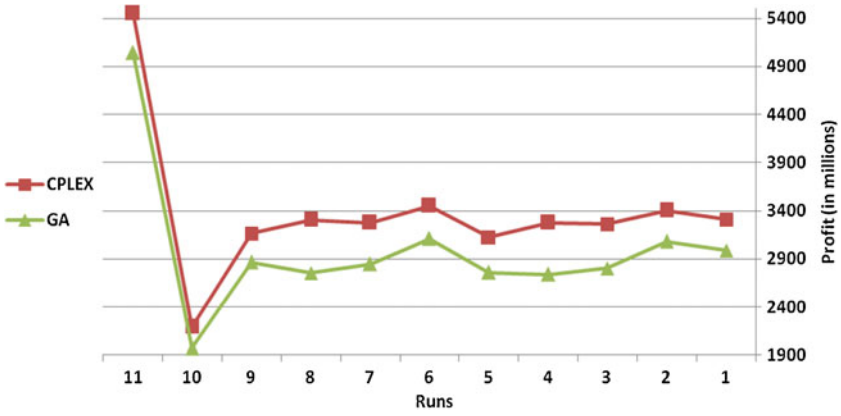

Fig. 5 Results of 11 runs (in millions)

optimum) and the genetic algorithm are recorded. The ranges of all parameters are demonstrated in Table 1. All algorithms achieve their best solution in a reasonable time. Results are illustrated in Table 3 and Fig. 5.

In analyzing the results of Table 3 and Fig. 5, we can conclude the following points:

- Distinctions between global optimums of CPLEX software and the results of the proposed genetic algorithm are low. The results show that just 11 % difference from the global optimum is admissible to persuade the acceptability of a proposed algorithm's performance. In addition, the mean of genetic algorithm outputs (2,996) is in the 95 % confidence limit of the global optimum of CPLEXs.
- The parameters of those different 11 runs are generated randomly, and we have two special runs (number 10 and 11) to analyze the algorithm in special cases. The proposed GA is performed well in these cases.

Consequently, regarding acceptable performance of the proposed genetic algorithm, it passed the first step of the validation process in comparison with CPLEX global optimization. Unfortunately, in real-sized instances and in practical situations, CPLEX cannot achieve global optimum in a reasonable time. Therefore, we need other inexact methods like the proposed GA. Hence, we experiment the developed meta-heuristic algorithm in large- and real-scale instances for our massive problem.

5.2 Large-sized instances analysis

One of the objectives of this paper is to evaluate the proposed genetic algorithm in large-scale instances. In the previous section, we could guarantee the acceptability of the results for the proposed algorithm. Now, 12 different sizes of instances are prepared and presented in Table 4. The characteristics of all instances are clarified in Table 4.

Table 4 characteristics of all instances

Instance	Number of							
	GA population	Suppliers, warehouses	Distributors, redistributors	Manufacturers	Disposal, disassembly centers	Customers, second customers	Periods	Products
1.	20 and 40	8	8	4	4	8	12	3
2.	20 and 40	10	10	5	5	10	12	3
3.	20 and 40	15	15	6	6	15	12	3
4.	20 and 40	20	20	7	7	20	12	4
5.	20 and 40	25	25	8	8	25	12	4
6.	20 and 40	30	30	9	9	30	12	4
7.	20 and 40	35	35	10	10	35	12	5
8.	20 and 40	40	40	11	11	40	12	5
9.	20 and 40	45	45	12	12	45	12	5
10.	20 and 40	50	50	13	13	50	12	6
11.	20 and 40	55	55	14	14	55	12	6
12.	20 and 40	60	60	15	15	60	12	6

Table 5 Results of large-scale analysis

Instance	Profits of (millions)		Difference of GA (20) and GA (40)	Time in minutes		Number of decision variables
	GA (20)	GA (40)		GA (20)	GA (40)	
1.	1,041	1,006	−3 %	1.6	1.9	812
2.	1,438	1,438	0 %	1.8	2.2	1,255
3.	1,864	1,870	0 %	2	2.6	2,490
4.	3,781	3,784	0 %	2.5	3.8	4,139
5.	5,061	4,955	−2 %	3.1	6.3	6,202
6.	5,207	5,385	3 %	4.7	8.1	8,679
7.	8,353	8,425	1 %	6.8	12	11,570
8.	8,322	9,412	12 %	8.1	16	14,875
9.	9,770	8,979	−9 %	10.4	20	18,594
10.	11,843	13,158	10 %	13.1	25	22,727
11.	10,301	9,382	−10 %	12.2	23	27,274
12.	12,427	15,409	19 %	14	27	32,235
Average	6,617	6,933	1.7 %	6.7	12.3	12,571

Reviewing Table 4, lead persuades us of many instances of those rarely seen in previous papers. We evaluate the proposed genetic algorithm in large-size instances. We deal with two genetic algorithm settings: population size "20" and "40." The results are illustrated in Table 5 and Figs. 6 and 7.

Analyzing the results of Table 5, Fig. 6, and Fig. 7, we conclude the following:

- Both proposed GAs could achieve the results in a reasonable time. The average time of obtaining results and finishing all iterations are 6.7 and 12.3 min for GA with population size 20 (GA 20) and GA with population size 40 (GA 40), respectively. These times are admissible in such cases.
- Interestingly, the results of two GAs (GA 20 and GA 40) are close to each other. The average differences of two GAs are just 1.7 %. These negligible differences lead us to conclude about the stability of the proposed GA, correctness of the parameter setting procedure, and the efficiency of results. Consequently, we cannot find any significant difference by reduplicating the population size. It should be pointed out that doubling the population size leads to achieve similar results in double time. It means that even if we duplicate the time of searching for an algorithm, the results will not improve more than 1.7 %. Finally, we trust the proposed GA in large-scale instances and in practical situations.
- Analyzing the number of decision variables and clarifying the vast problem we cope with and the value of algorithms that can give us acceptable results. As an example, in instance 12, there are around 32 thousand decision variables including 285 binary variables. Therefore, if we try to estimate just assigning these 285 binary variables, we would have 2^{285} different. Therefore, we need an acceptable and well-performing algorithm for such problems.

As analyzed, the performance of the proposed genetic algorithm is highly acceptable in small- and large-scale instances. Roughly speaking, we can trust the proposed genetic algorithm in the design and planning of a closed-loop

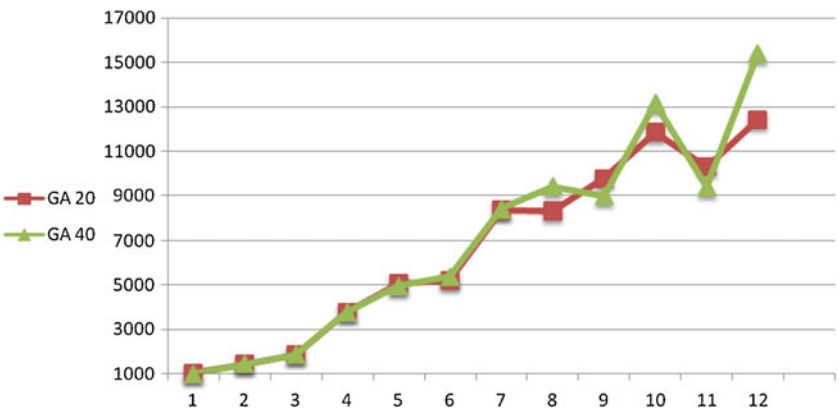

Fig. 6 Results of large-scale analysis

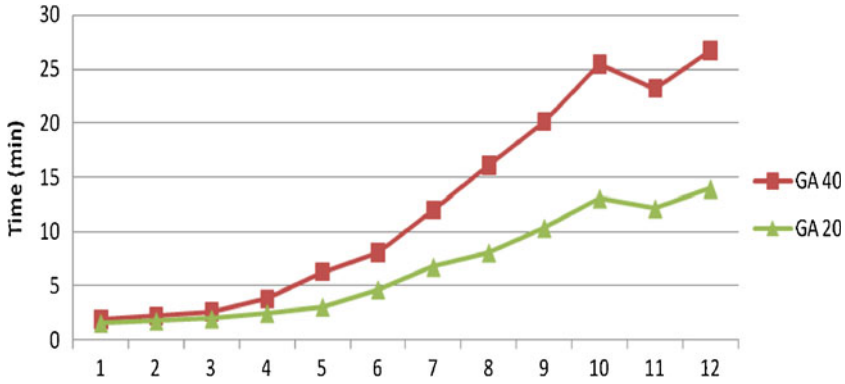

Fig. 7 Time analysis of the results of large-scale analysis

supply chain problem. It achieves an acceptable solution in a very reasonable time.

6 Conclusion and future researches

This paper copes with a very important problem about designing and planning a closed-loop supply chain. Based on analyzing current gaps in literature, a necessary comprehensive multi-echelon multi-period and multi-product model is proposed. This model contains almost all possible entities and flows in CLSC network.

On the other side, despite some claims, few papers consider large-scale instances in the field of closed-loop supply chain. Therefore, an appropriate genetic algorithm is proposed to solve different size of instances. A complete computational analysis is performed to validate the genetic algorithm and set its parameters. First, the model is coded by IBM ILOG CPLEX 12.2 optimization software to achieve the global optimum. Meanwhile, a genetic algorithm is coded by Matlab 7.12 software to complete comparing. Secondly, 11 runs of a small-sized instance are developed, and performance of the genetic algorithm is compared with the global optimum of CPLEX. On average, the results show just 11 % distance to the global optimum. Third, in order to cope with large-scale instances, 12 different sizes of instances including very large-scale ones (with 356,000 decision variables) are generated. Then, two developed GAs are compared with each other. They both perform very well with a difference of just 1.7 %. The results prove the constancy of the proposed genetic algorithm and its applicability in real-size situations. On average, they could achieve the best solutions in “6.7” and “12.3”min for GA-20 and GA-40, respectively. It is a reasonable and acceptable time for such instances.

However, there is some guidance for future research; first, we suggest evaluating the proposed genetic algorithm in a practical environment to modify it and to clarify its shortcomings. Second, the genetic algorithms could be compared with other meta-heuristics like PSO. Finally, the deterministic approach of this paper could be improved by considering nondeterministic parameters.

References

1. Chopra S, Meindl P (2007) Supply chain management: strategy, planning and operation, 3rd edn. Pearson Prentice Hall Inc., New Jersey, ISBN 81-7758-003-5
2. Beamon BM (1999) Designing the green supply chain. Logist Inf Manag 12(4):332–342
3. Fleischmann M, Krikke HR, Dekker R, Flapper SDP (2000) A characterisation of logistics networks for product recovery. Omega 28:653–666
4. Schultmann F, Zumkeller M, Rentz O (2006) Modeling reverse logistic tasks within closed-loop supply chains: an example from the automotive industry. J Oper Res 171(3):1033–1050
5. Pibernik R, Sucky E (2006) Centralised and decentralised supply chain planning. Int J Integ Supply Manag 2(1/2):6–27
6. French ML, LaForge RL (2006) Closed-loop supply chains in process industries: an empirical study of producer re-use issues. J Oper Manag 24(3):271–286
7. Prahinski C, Kocabasoglu C (2006) Empirical research opportunities in reverse supply chains. Omega 34(6):519–532
8. Biehl M, Prater E, Realff MJ (2007) Assessing performance and uncertainty in developing carpet reverse logistics systems. Comput Oper Res 34(2):443–463
9. Kusumastuti RD, Piplani R, Lim GH (2008) Redesigning closed-loop service network at a computer manufacturer: a case study. Int J Prod Econ 111(2):244–260
10. Lee DH, Dong M (2008) A heuristic approach to logistics network design for end-of-lease computer products recovery. Transp Res E Logist Transp Rev 44(3):455–474
11. (de la) Fuente MV, Ros L, Cardós M (2008) Integrating forward and reverse supply chains: application to a metal-mechanic company. Int J Prod Econ 111(2):782–792
12. Amaro ACS, Barbosa-Póvoa AFPD (2008) Planning and scheduling of industrial supply chains with reverse flows: a real pharmaceutical case study. Comput Chem Eng 32(11):2606–2625
13. Mutha A, Pokharel S (2009) Strategic network design for reverse logistics and remanufacturing using new and old product modules. Comput Ind Eng 56(1):334–346
14. Dehghanian F, Mansour S (2009) Designing sustainable recovery network of end-of-life products using genetic algorithm. Resour Conserv Recycl 53(10):559–570

15. Xanthopoulos A, Iakovou E (2009) On the optimal design of the disassembly and recovery processes. Waste Manage 29(5):1702–1711
16. Kannan G, Noorul Haq A, Devika M (2009) Analysis of closed loop supply chain using genetic algorithm and particle swarm optimisation. Int J Prod Res 47(5):1175–1200
17. Pokharel S, Mutha A (2009) Perspectives in reverse logistics: a review. Rev Article Resour Conserv Recycl 53(4):175–182
18. Subramoniam R, Huisingh D, Chinnam RB (2009) Remanufacturing for the automotive after market- strategic factors: literature review and future research needs. J Clean Prod 17(13):1163–1174
19. Wang HF, Hsu HW (2010) A closed-loop logistic model with a spanning tree based genetic algorithm. Comput Oper Res 37(2):376–389
20. El-Sayed M, Afia N, El-Kharbotly A (2010) A stochastic model for forward–reverse logistics network design under risk. Comput Ind Eng 58:423–431
21. Gold S, Seuring S, Beske P (2010) The constructs of sustainable supply chain management—a content analysis based on published case studies. Prog Ind Ecol 7(2):114–137
22. Subramanian P, Ramkumar N, Narendran T (2010) Mathematical model for multi-echelon, multi-product, single time-period closed loop supply chain. Int J Bus Perform Supply Chain Model 2(3/4):216–236
23. Kannan G, Sasikumar P, Devika K (2010) A genetic algorithm approach for solving a closed loop supply chain model: a case of battery recycling. Appl Math Model 34:655–670
24. Pishvaee MS, Rabbani M, Torabi SA (2011) A robust optimization approach to closed-loop supply chain network design under uncertainty. Appl Math Model 35:637–649
25. El korchi A, Millet D (2011) Designing a sustainable reverse logistics channel: the 18 generic structures framework. J Clean Prod 19(6–7):588–597
26. Gomes Salema MI, Barbosa-Povoa AP, Novais AQ (2010) Simultaneous design and planning of supply chains with reverse flows: a generic modelling framework. Eur J Oper Res 203(2):336–349
27. Amaro ACS, Barbosa-Póvoa AFPD (2009) The effect of uncertainty on the optimal closed-loop supply chain planning under different partnerships structure. Comput Chem Eng 33(12):2144–2158
28. Lieckens K, Vandaele N (2007) Reverse logistics network design with stochastic lead times. Comput Oper Res 34(2):395–416
29. Pishvaee MS, Zanjirani Farahani R, Dullaert W (2010) A memetic algorithm for bi-objective integrated forward/reverse logistics network design. Comput Oper Res 37(6):1100–1112
30. Min H, Ko HJ (2008) The dynamic design of a reverse logistics network from the perspective of third-party logistics service providers. Int J Prod Econ 113(1):176–192
31. Holland JH (1992) Adaptation in natural and artificial systems, 2nd edn. MIT Press, Cambridge
32. Hartl RE (1990) A global convergence proof for a class of genetic algorithms. Working paper, Vienna University of Technology, Institute of Biometrics, Operations Research and System Theory
33. Radcliffe NJ (1993) Genetic set recombination. Edinburgh Parallel Computing Centre, University of Edinburgh
34. Bean JC (1994) Genetics and random keys for sequencing and optimization. INFORMS J Comput 6(2):154–160
35. Miller BL, Goldberg DE (1996) Genetic algorithms, tournament selection, and the effects of noise. Evol Comput 4(2):113–131
36. Vose MD (1991) Generalizing the notion of schema in genetic algorithms. Artif Intell 50(1):385–396
37. Koza JR (1991) Genetic programming: II: automatic discovery of reusable programs. MIT Press, Cambridge
38. Reeves CR (1993) Modern heuristic techniques for combinatorial problems. Blackwell Scientific Publications, Oxford
39. Mitchell M (1996) An introduction to genetic algorithms. MIT Press, Cambridge
40. Baeck T, Fogel DB, Michalewicz Z (1997) Handbook of evolutionary computation. Oxford University Press/Institute of Physics Publishing, New York