Applied Mathematical Modelling 39 (2015) 3990–4012

Contents lists available at ScienceDirect

Applied Mathematical Modelling

journal homepage: www.elsevier.com/locate/apm

A hybrid particle swarm optimization and genetic algorithm for closed-loop supply chain network design in large-scale networks

Hamed Soleimani [a,*], Govindan Kannan [b]

[a] Faculty of Industrial and Mechanical Engineering, Qazvin Branch, Islamic Azad University (IAU), Qazvin, Iran
[b] Department of Business and Economics, University of Southern Denmark, Odense, Denmark

ARTICLE INFO

Article history:
Received 30 September 2012
Received in revised form 18 November 2014
Accepted 11 December 2014
Available online 19 December 2014

Keywords:
Closed-loop supply chain
Genetic algorithm
Particle swarm optimization
Hybrid algorithm

ABSTRACT

Today, tracking the growing interest in closed-loop supply chain shown by both practitioners and academia is easily possible. There are many factors, which transform closed-loop supply chain issues into a unique and vital subject in supply chain management, such as environmental legislation, customer awareness, and the economical motivations of the organizations. However, designing and planning a closed-loop supply chain is an NP-hard problem, which makes it difficult to achieve acceptable results in a reasonable time. In this paper, we try to cope with this problem by proposing a new and effective solution methodology. On the other hand, this research considers improving closed-loop supply chain network optimization processes through dealing with mathematical programming tools; developing a deterministic multi-product, multi-echelon, multi-period model; and finally presenting an appropriate methodology to solve various sizes of instances. Both design and planning decision variables (location and allocation) are considered in the proposed network. Besides, in order to have a reliable performance evaluation process, large-scale instances are regarded in computational analysis. Two popular meta-heuristic algorithms are considered to develop a new elevated hybrid algorithm: the genetic algorithm (GA) and particle swarm optimization (PSO). Analyzing the above-mentioned algorithms' strengths and weaknesses leads us to attempt to improve the GA using some aspects of PSO. Therefore, a new hybrid algorithm is proposed and a complete validation process is undertaken using CPLEX and MATLAB software. In small instances, the global optimum points of CPLEX for the proposed hybrid algorithm are compared to genetic algorithm, and particle swarm optimization. Then, in small, mid, and large-size instances, performances of the proposed meta-heuristics are analyzed and evaluated. Finally, a case study involving an Iranian hospital furniture manufacturer is used to evaluate the proposed solution approach. The results reveal the superiority of the proposed hybrid algorithm when compared to the GA and PSO.

* Corresponding author.
E-mail addresses: h.soleimani@qiau.ac.ir, hd_soleimani@yahoo.com (H. Soleimani).

http://dx.doi.org/10.1016/j.apm.2014.12.016
0307-904X/

1. Introduction

The closed-loop supply chain (CLSC) has changed from being an undesirable constraint into being an economically acceptable necessity. Interestingly, it also has a close relationship with the sustainable supply chain [1]. In a classical supply chain (forward supply chain), the facilities attempt to realize the processes of fulfilling a customer's request. As customers do not use products forever, the supply chain's responsibilities also continue at the end of the products' life cycle. Reverse supply chain deals with the above-mentioned issues. Initially, some disassembly centers collect end-of-life products from customers, which are called "return products" in our context. Afterwards, some appropriate decisions are made: Some return products need some repairs after which they are qualified to be sold in the second-hand markets. They are usually repaired in disassembly centers and then distributed through redistributors. Some return products have to be forwarded to the manufacturers to be remanufactured and then sold to the second customers. Some can be used partially as raw materials, and hence they are sent to the suppliers for recycling. Finally, those parts which cannot be reused are forwarded to disposal centers (environment-friendly disposal units). A CLSC is achieved when forward/reverse processes are considered simultaneously.

An illustration of a simple CLSC is presented in Fig. 1 by Beamon [2]. Disposal centers are demonstrated by "W" through a supply chain network. Forward network flows are illustrated by a solid line and reverse flows by a dashed line.

Generally, in dealing with a CLSC in more than a short term, bi-level important decisions should be made: Design and planning. In the designing stage, which is a long-term analysis, strategic decisions are undertaken on the main characteristics of all facilities, such as network configurations, structure, capacities, coordination, etc. Then an important part of the supply chain network including the quantity of flows between all supply chain network entities should be determined at the planning level [3]. In this study, network configurations are considered in the designing stage (also called "location" here). They can clarify whether a facility should be activated (founded) in the final network or not. Besides, network flows are determined in the planning stage, which are also called allocation decisions here.

On the other hand, dealing with the design and planning problem of a CLSC as an NP-hard problem requires effective solution approaches, which can give us reliable solutions in a reasonable time especially for real-sized problems. Earlier publications, which tried to propose new solution methodologies, can give some evidence about this necessity. Besides, studying the small-size instances in the earlier research ensures that we evaluate solution methodologies in various sizes of instances. It should be mentioned that without availability of effective solution approaches, it would be difficult to apply such optimization models to real-world problems. In fact, more attempts are necessary to convert such theoretical models into practice. The missing part is engendered due to the inefficiencies encountered when using current methods to solve large-scale problems. Roughly, as CLSC design and planning is an NP-hard problem [4,5], proposing well-behaved methodologies can help practitioners to cope with the mentioned problem by achieving acceptable solutions in a reasonable time.

Finally, the aim of this paper is to cope with a large and complex CLSC design and planning problem through a deterministic approach where using meta-heuristic algorithms is an appropriate approach. The term "complex" for this problem refers to its Non-deterministic Polynomial (NP-hard) characteristics [4,5] and also to the size of the evaluated instances. Therefore, this study deals with a deterministic, large scale CLSC design and planning problem and the aim is to provide successful tools for solving large-scale problems. In addition, here, when referring to complexity issues we mean the structural

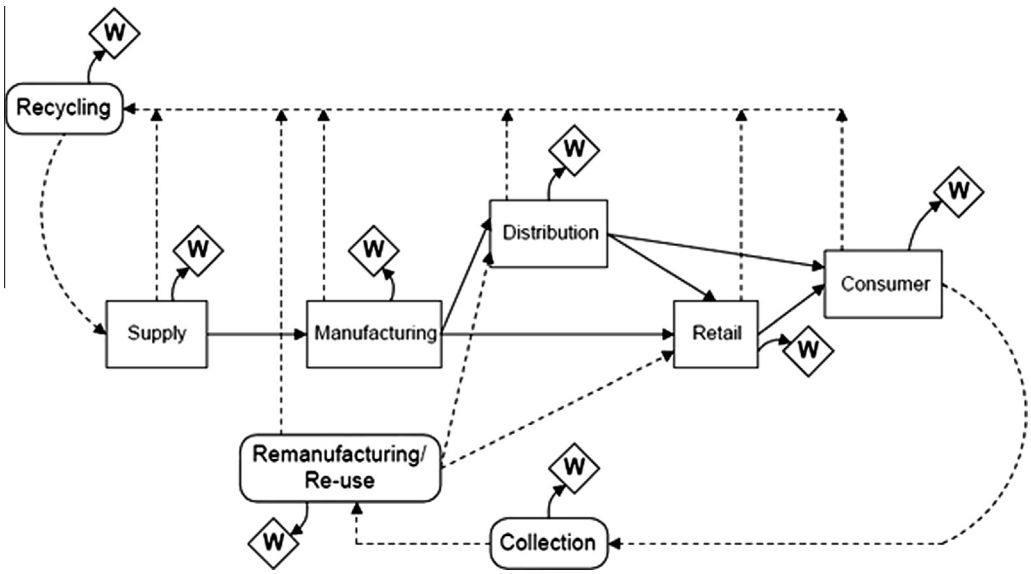

Fig. 1. A general closed-loop supply chain [2].

complexity (size) and not functional complexity (stochastic and uncertainty issues). Indeed, as we will show, the optimum point can be achieved in small-size instances but not in a large one. In order to cover this aim, a newly developed hybrid algorithm is proposed to deal with these issues. The proposed algorithm is based on the genetic algorithm (GA), which is improved utilizing the strengths of particle swarm optimization algorithm. A complete validation of the proposed algorithm is undertaken through the usage of different sizes of instances and analyzing a real case study involving a large hospital furniture manufacturer.

The rest of this paper is organized as follows: In Section 2, a complete literature review is discussed in order to clarify the current gaps. A description of the model and its formulation is presented in Section 3. Section 4 concludes the complete explanation of the proposed hybrid algorithm. A comprehensive validation process of the proposed algorithm in small and large-scale instances is performed in Section 5 through comparing the hybrid's solutions with CPLEX global optimum and the results of particle swarm optimization (PSO), and GA algorithms. Sensitivity analyses of fixed costs are developed in Section 6. Section 7 is assigned to managerial implications and case study investigation. Finally, conclusion, and suggestions for future research are provided in Section 8.

2. Literature review

In this paper, we attempt to propose a new and elevated solution methodology to deal with CLSC network design and planning, which is an important concern of supply chain researchers. Since this study has two main goals: Dealing with the CLSC problem and developing a new solution methodology. Consequently, the focus of the literature survey in this study, is on the CLSC design problems and on analyzing their solutions. Large-size instances, especially those using metaheuristic algorithms, are the main issue here.

There are various studies dealing with CLSC design and planning (mostly published before 2010), which consider different aspects of network design problems. Generally, these studies develop modeling characteristics and mostly utilize general exact solvers or exact algorithms such as branch and bound, branch and price, etc. Gomes Salema et al. in [6,7] employed a CLSC design and planning problem with some case studies where GAMS/CPLEX software was used as a solver. Amaro and Barbosa-Póvoa [8] in continuation of the earlier pharmaceutical supply chain study in Amaro and Barbosa-Póvoa [9], proposed a closed-loop problem, which is an analysis of continuous and discrete formulation models using the standard branch and bound method. However, it did not cover designing-stage decisions.

It should be mentioned here that there are many papers with different characteristics on designing and planning a CLSC but with small case studies or computational analysis. Examples include Kusumastuti et al. [10] using LINGO software, Zhou and Wang [11] and Amaro and Barbosa-Póvoa [12] utilizing the branch and bound method, Mutha and Pokharel [13] and Gomes Salema et al. [14] exploiting GAMS software, Pishvaee et al. [15] utilizing LINDO software, Li and Marlin [16] using interior point method, and El-Sayed et al. [17] applying CPLEX software to solve small-sized problems.

On the other hand, applying simulation-based optimization techniques is another approach to dealing with CLSC design and planning, which generally considers stochastic situations. The attempt of this paper is to develop a new methodology for the same problem but in deterministic situations. Some of the related publications in this area in which simulated best optimization approaches are utilized should be mentioned here: Georgiadis [18] considered a single-producer closed-loop recycling network that tries to use recycled materials as its principal inputs of raw materials. It proposed a new decision-making method for capacity planning based on the dynamic periodic review approach and balanced tradeoff between profit and capacity utilization. The presented methodology tried to integrate capacity planning into a system dynamics model. Besiou et al. [19] studied the impact of scavenging on the operations of a Waste Electrical and Electronic Equipment (WEEE) recovery system through the environmental dimension (availability of natural resources and landfills, and pollution due to leaks from the scavengers' uncontrollable disposal ignoring pollution due to emissions), the economic dimension (CLSC profitability), and the social dimension of sustainability (number of unemployed scavengers). Then they assessed the efficiency of different regulatory measures in limiting the impact of informal recycling on the economic, environmental, and social aspects of sustainability. Such studies consider stochastic parameters; however, we try to develop a new solution methodology for a deterministic model. Georgiadis and Athanasiou [20] extended the system dynamic-based simulation model by Georgiadis et al. [21]. They considered two types of product and their lifecycles under two market scenarios regarding the customer preferences. Recently, they have dealt with strategic demand-driven policies of capacity planning in a reverse logistics network with remanufacturing. They also considered high capacity acquisition cost and uncertainty in demand, sales patterns, quality, and timing of end-of-use product returns [22].

In order to cover more related papers, some heuristic and metaheuristic studies will be reviewed in detail and then a conclusion table is presented. This literature review reveals successful applications of metaheuristic algorithms and high acceptability of GA and PSO. Kannan et al. [23] proposed a genetic algorithm and particle swarm optimization to design and plan a CLSC which both displayed acceptable performance when dealing with real problems. He in 2010, he continued his works in planning CLSC utilizing genetic algorithm, where he validated this popular metaheuristic algorithm in small-sized instances [24]. Such research motivates us in utilizing GA and PSO and elevating their performances. Researchers have utilized the genetic algorithm frequently because of its excellent performance in CLSC design and planning. Hence, it was chosen as the backbone of the proposed hybrid algorithm. Some GA-based studies should be mentioned here: Wang and Hsu [25] utilized a revised spanning-tree-based genetic algorithm developed by using determinant encoding representation for solving

this NP model. The reverse chain of the above-mentioned model was so simple that there was no validation or usage of real/large-scale instances (just 3 suppliers and 5 manufacturers). Dehghanian and Mansour [26] presented a large problem investigation, which was solved by a multi-objective GA method but in reverse logistic and in planning level. Pishvaee et al. [27] studied a single period, single products problem using a memetic algorithm. Other heuristic algorithms are also utilized in this field: Lee and Dong [28] proposed a two-stage heuristic methodology that decomposes logistics network design into a location–allocation problem in computer industry. A tabu search (TS) algorithm is developed in the second stage to provide an elevated transportation solution for returned products. The computational analysis is fairly large (up to 40 facilities and 100 customers).

Reviewing related papers leads to noteworthy works. In summary, Table 1 is prepared as a comprehensive conclusion of the whole literature review. The aim of this paper and its characteristics are clarified in the last row, where the focus is on solving real instances for both design and planning decisions of a CLSC problem.

A precise analysis of Table 1 leads to some important points about the necessity of this research. Most of the publications of Table 1 present various design/planning studies in which authors regard modeling characteristics and the solving of small instances utilizing general exact solvers [33,15,6,22,29,30,42–44,14,31,32,34]. A few researchers try to develop exact algorithms such as branch and bound, interior point method, and analytical approaches in order to cope with design and planning a CLSC [16,35,12,36,8]. Generally, due to the limitations of exact solvers and approaches, such algorithms cannot solve large-size instances.

On the other hand, since using metaheuristic algorithms is the most reasonable approach in dealing with such NP-hard problems, there are valuable studies in this area. The other publications of Table 1 illustrate the CLSC design and planning research that utilizes various metaheuristics such as GA [24,25,45], GA and PSO [23], tabu search [28], memetic [27], and hybrid PSO–GA [46]. These studies reveal the acceptability of metaheuristic algorithms and their potential for being evolved. However, most of them present simple evaluation procedures through simple and small instances, which could not convince readers about their applicability. In this paper, we try to elevate the trend of solution methodology evolution by proposing a new hybrid solution approach and analyze its robustness through a comprehensive evaluation study. It should be mentioned

Table 1
Literature review of closed-loop supply chain design and planning.

Row	References	Year	Period	Product	Exact approach	Size of instance	Meta heuristics	Decision level
1	Özkir and Basligil [29]	2013	Multi	Single	GAMS	Small	–	Design and planning
2	Özceylan and Paksoy [30]	2013	Multi	Single (Multi part)	CPLEX	Small	–	Design and Planning
3	Ramezani et al. [31]	2013	Single	Multi	–	Small	–	Design, planning, and capacity
4	Amin and Zhang [32]	2012	Single	Multi	GAMS	Small	–	Design and planning and fuzzy suppliers selection
5	Wang et al. [33]	2011	Single	Single	GAMS	Small	–	Planning
6	Pishvaee et al. [15]	2009	Single	Single	LINGO	Small	–	Design and planning
7	Gomes Salema et al. [6]	2006	Single	Multi	CPLEX	Small	–	Design and planning
8	El-Sayed et al. [17]	2010	Multi	Single	CPLEX	Small	–	Design and planning
9	Paksoy et al. [34]	2011	Single	Multi	LINDO	Small	–	Planning and transportation mode
10	Qiang et al. [35]	2013	Single	Single	Analytical		–	Planning
11	Khajavi et al. [36]	2011	Single	Single	B&B using LINGO	Small	–	Design, planning, and capacity
12	Özceylan and Paksoy [42]	2013	Multi	Single (Multi part)	CPLEX	Small	–	Design and planning
13	Wang et al. [43]	2013	Single	Single	CPLEX	Small	–	Design and planning
14	Metta and Badurdeen [44]	2011	Single	Multi	CPLEX	Small	–	Design and planning
15	Salema and Barbosa-Povoa [14]	2010	Multi	Multi	GAMS	Small	–	Design and planning
16	Li and Marlin [16]	2009	Single	Single	Interior point	Small	–	Planning
17	Amaro and Barbosa-Póvoa [12]	2009	Multi	Single	B&B	Small	–	Design and planning
18	Amaro and Barbosa-Póvoa [8]	2007	Single	Multi	B&B	Mid-size	–	Planning
19	Kannan et al. [24]	2010	Multi	Multi	GAMS	Small	GA	Planning
20	Wang and Hsu [25]	2010	Single	Single	CPLEX	Small	GA	Design and planning
21	Soleimani et al. [45]	2013	Multi	Multi	CPLEX	Large	GA	Design and planning
22	Kannan et al. [23]	2009	Single	Multi	No	Small	GA and PSO	Planning
23	Pishvaee et al. [27]	2010	Single	Single	No	Large	Memetic	Design and planning
24	Lee and Dong [28]	2008	Single	Multi	No	Large	Tabu Search	Design and planning
25	Zhou et al. [46]	2012	Single	Multi	–	Small	Hybrid PSO–GA	Design
26	Our research	2014	Multi	Multi	CPLEX	Large	Hybrid GA–PSO	Design and planning

that the study of Zhou et al. [46] is a single-period analysis in order to determine designing stages (and some assignments of allocation, but not flows), so their hybrid is a simple one with a small numerical example. Roughly, they just try to design binary decision variables. In this study, on the other hand, all flow decision variables are determined in a multi-period study through a complete validation procedure. The proposed hybrid in [46] is based on PSO with GA operators, while this paper proposed a GA based metaheuristic with PSO operators.

From the stock of other recent related publications, two papers should be discussed here: Amin and Zhang [47] developed a cost-benefit analysis for a reuse and recycling facility. They evaluate reuse and recycling costs and benefits and also operational and investment costs in order to investigate the profitability of the reuse and recycling facility. They propose a mixed-integer linear programming model that minimizes the total cost. They develop a multi-product network with just 4 echelons: Plants, collection centers, disposal centers, and customers. They investigate the impact of demand and return uncertainties on the network configuration by stochastic programming (scenario-based). Besides, Choi et al. [48] examine different channel leadership in CLSC. They tried to compare the retailer-Stackelberg model with the manufacturer-Stackelberg model.

Finally, the following gaps are determined from the analyses of Table 1:

- From the modeling point of view, the lack of an appropriate multi-product, multi-period model for designing and planning a CLSC network is clarified. Indeed, when products and periods are considered, it leads to just three papers [14,24,45]. Salema and Barbosa-Povoa [14] used a graph approach based on conventional concepts of nodes and arcs to develop a generic model for simultaneous design/planning of supply chains with reverse flows. Two other papers consider pure GA [45] or PSO [24] algorithms. Indeed, in [45] the authors try to present a comprehensive model (as they mention explicitly), not a new solution methodology, so they just develop a GA. Therefore, their study is just an application of GA in a CLSC network with a new and complete formulation, but the analysis of the literature clarifies the necessity for new and efficient solution methodologies in order to cope with real huge networks. Therefore, in this study we use the same model with some modifications in flows and assumptions, and we focus on developing solution approaches. Besides, the largest network in [45] contains 60 suppliers and 15 manufacturers, which is interestingly equal to the smallest network in this study. Indeed, the number of instances in this study is drastically larger than in [45] which includes 12,571 decision variables in each instance on average, but in this study the average number of decision variables is 113,022, which is highly larger than [45]. Further, Özceylan and Paksoy [30,42] also considered multi-period, multi-part networks. The first paper tried to propose a new mixed integer model for a CLSC multi-period, multi-part network in a location allocation problem that includes both forward and reverse flows. The latter is a fuzzy multi-objective model which used the fuzziness in the capacity, objectives, demand constraints, and reverse rates. Consequently, a multi-period, multi product model, which considers all possible flows in its network, is considered to make it close to real situations.
- Important gaps are considered in terms of solution methodologies. Since CLSC design and planning is an NP-hard problem, general exact solvers cannot solve real-size instances within an acceptable time. Some attempts were made to develop meta-heuristic algorithms such GA, PSO, TS, and memetic algorithm. However, developing new methodologies which are capable of achieving better results regarding solution quality and time, are required to ensure higher profits and lower costs. Thus, this paper develops a new PSO–GA hybrid method to elevate capabilities of solving complex problems.

Finally, a multi-product, multi-period, mixed-integer programming model is developed in this paper, which considers different possible flows of real situations between network entities. Besides, in order to elevate solving capabilities, this paper proposes a new hybrid algorithm.

3. Model description and formulation

A model with sufficient capabilities is proposed to come up with real problems. The model is based on [45] with some differences in flows and assumptions. There are two main modifications in flows here:

a. As it can be seen in Fig. 1 in [45], there is no flow between manufacturers and distributors. However, it could be possible for a manufacturer to ignore warehouses and send the products to the distributors directly. It can lead to better solutions. We add this possibility to the network and also the appropriate modifications to the formulation.
b. As it can be seen in constraint (14) and (21) in [45], the disassembly centers cannot send the recovered products to the second customers directly. It means, there is no direct-shipment alternative for collection centers. However, in some cases, with small recovery processes, we can send the return products to the second customers. We add this option here and the model is a bit different in terms of formulations.

The following are some assumptions that form the model:

- The model is multi-echelon, multi-product, and multi-period and consists of these echelons: suppliers, manufacturers, warehouses, distributors, retailers (customers), disassembly centers (collection centers), redistributors, disposal centers, and second customers (see Fig. 2).

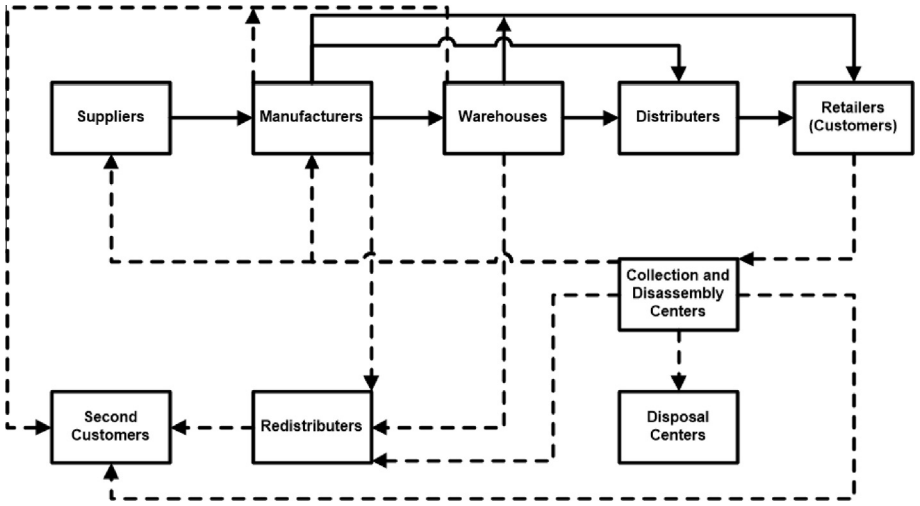

Fig. 2. The modified model (The *arrows* show the possible flows).

- The quality, demand, and prices of returned products are different from those of first products and they cannot be sold interchangeably.
- The potential locations, capacities of all facilities, and cost parameters are predetermined.
- Shortages cannot be covered in the next periods and if customers' demands cannot be met, a shortage cost should be endured. On the other hand, in terms of shortage assumption, we add a minimum service level requirement to the model. Actually, we accept the shortage option for the model, but the solutions should cover at least 70% of customers' demand. Therefore, the minimum service level is predetermined here in order to be closer to the real situation (no service level requirement in [45]).

All flows among network entities are presented in Fig. 2. It should be mentioned that forward and reverse flows are illustrated by solid lines and dashed lines, respectively.

Undoubtedly, formulating a conceptual model is the first step toward making it practical. Some differences caused by generality purpose ensure that the proposed model is based on the notation of [17,45]. Here, an overview of the main parts of the model is explained and the details of the formulation are presented in Appendix A.

The main decision variables are location and allocation variables. Here, L_i is the 0–1 variable equals "1" if location "i" is open and "0" otherwise. Besides, Q_{ijut} is the flow of batches of product "u" from node "i" to node "j" in period "t". On the other hand, we have considered profit as the objective function, so we should calculate all sales and costs as follows:

First products sale (flows from distributors, manufacturers, and warehouses to the customers):

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

Second products sale (flows from redistributors, manufacturers, and warehouses to the second customers):

$$\sum_{r\in R}\sum_{k\in K}\sum_{u\in U}\sum_{t\in T} Q_{rkut}B_{ru}P_{kut} + \sum_{f\in F}\sum_{k\in K}\sum_{u\in U}\sum_{t\in T} Q_{fkut}B_{fu}P_{kut} + \sum_{w\in W}\sum_{k\in K}\sum_{u\in U}\sum_{t\in T} Q_{wkut}B_{wu}P_{kut}. \tag{2}$$

Here, B_{iu} explain the batch size in entity i for product u. Besides, P_{cut} and P_{kut} are unit price of product "u" at the first customer "c" in period "t", and unit price of product "u" at the second customer "k" in period "t" respectively.

Further, total cost should be calculated including various cost including fixed opening costs, material supplying costs, manufacturing costs, non-utilized capacity costs, shortage costs, purchasing costs of return products from customers, disassembly costs of return products, recycling costs, remanufacturing costs, repairing costs, disposal costs, transportation costs, and inventory holding costs.

The details of cost calculations are presented in Appendix A.

The constraints of the model are divided in four groups: balanced constraints, capacity constraints, linking and shipping constraints, and finally, maximum-number of allowable-location constraints.

Balance constraints ensure the equilibrium between entering flows of a node and its issues. At each node, all entering flows of every product per period should be equal to all issuing flows from that node. For all the entities in Fig. 2, this constraint should be seen. Capacity constraints control maximum flows that can enter into or issue from each entity (facility). Linking or shipping constraints consider the possible/impossible flows between entities. Such constraints guarantee no links

between nodes without actual real flows and no flows between two nodes sans an actual link. Finally, since there are limitations on the number of activated locations, we need some constraints for the maximum number of allowable locations.

The complete model is explained in the Appendix A.

4. The proposed hybrid algorithm

As stated, traditional techniques are inefficient when practical search space is too large (like this problem). For instance, here, a network with only five units in each entity will realize a problem with 14,482 constraints and 11,336 decision variables including 460 binary variables. Hence, there is a huge search space for this small instance.

Based on studies of this research, a hybrid of two popular and well-behaved algorithms is proposed to ensure a more desirable algorithm for large-scale problems. Both the chosen metaheuristic algorithms can meet expectations and have demonstrated satisfactory performances in earlier studies. Hence, this is the reason for creating a new hybrid algorithm containing the positive points of both.

4.1. Particle swarm optimization and genetic algorithm

PSO is a recent evolutionary algorithm, developed by Kennedy and Eberhart [37], which had its genesis from trying to replicate bird flocking, fish schooling, and swarm theory. Similar to the GA, PSO is a population-based optimization tool (called particles), which has fitness values to evaluate the population [23]. It can also update the population to achieve an optimal solution. Similar to the GA, PSO does not guarantee achieving the optimum point [23]. In an iteration, PSO remembers the best solution between populations (“gbest”) and the best solution for each population (“pbest”). Preserving gbest and pbest is very useful characteristics of PSO, which the GA lacks. Hence, the proposed hybrid algorithm has evolved a GA by trying to preserve pbest and gbest, and to utilize them in offspring procedures.

On the other hand, GA is a highly popular metaheuristic optimization technique, originally developed by Holland [38] and based on the genetic procedure of the human body and on Darwin’s theory of the “survival of the fittest”. It was later applied to various optimization problems in different fields of science such as engineering, and operations research. In the 1990s, different aspects of the GA were extended; Hartl [39] researched the convergence of the algorithm, and Radcliffe [40] and Bean [41] studied the crossover operator (their studies are used here). Miller and Goldberg [49] studied the selection strategies, and Vose [50] and Koza [51] investigated the entire concept of the GA of Holland by proposing genetic programming. More information about genetic algorithm and other evolutionary algorithms can be found in Reeves [52], Mitchell [53], and Baeck’s et al. [54] studies.

A GA starts with a random population of solutions (called chromosomes) and then tries to improve solutions through many iterations called generations. Each solution’s performance is evaluated by a fitness function corresponding to the objective function of the optimization problem. Following parental selection, crossover and mutation operators are applied. Crossover combines materials from parents to produce children. On the other hand, mutation makes small local changes in feasible solutions to provide population diversity for a wider exploration of feasible solutions. The mutation operator is defined to ensure that generated solutions are not trapped in some local optima. As the final solution is independent of initial solutions, the basic population is randomly generated in most cases [23].

Table 2 presents analyses of the strengths and weaknesses of PSO and GA to provide a holistic view of both and also to clarify characteristics of the proposed hybrid algorithm.

As seen in Table 2, attempts are undertaken in order develop a GA by adding capability to preserve its best solution for each particle in all iterations (personal best or pbest) and the best result of all populations in all iterations (global best or gbest). The new capabilities are called PSO operators. Both “pbest” and “gbest” have an important role in the proposed algorithm’s evolutionary strategies. Such an algorithm can achieve more robust solutions in a reasonable time. Besides, since the proposed algorithm includes the advantages of both PSO and GA, the potential of achieving better objective functions is clearly higher than with GA and PSO individually. In order to prove such assumptions, a comprehensive evaluation procedure is undertaken in the following sections.

Table 2
Strengths and weaknesses of PSO, GA, and proposed hybrid algorithms.

Characteristics	Genetic algorithm	Particle swarm optimization	Proposed Hybrid algorithm
Random initial population	Yes	Yes	Yes
Evolutionary strategies	Yes	Yes	Yes
Crossover operator	Yes	No	Yes
Mutation operator	Yes	No	Yes
Preserving the best solution of all iteration (gbest)	No	Yes	Yes
Preserving the best result of every single solution (particle) (pbest)	No	Yes	Yes
Guarantee of achieving global optimum or any specific approximation	No	No	No
Predetermining the time of solving process	No	No	No

4.2. New hybrid algorithm

Based on the complementary points of GA and PSO, hybrids of these algorithms exist in other fields. They should necessarily be mentioned before presenting the proposed new hybrid algorithm to clarify its distinctions.

Guang-Min et al. [55] provided a GA to solve the linear bi-level programming, which divides all particles into two clusters based on the fitness value performance. The first cluster includes particles with the best fitness values while the second cluster has particles with the worst fitness values. Crossover and mutation implementation occurs only in the first cluster to update the fitness values of particles. Finally, all particle movement is based on PSO rules. Du et al. [56] utilized crossover and mutation operators of GA to update PSO particles. It was actually a PSO-based algorithm. Again, to solve bi-level linear programming, Kuo and Han [57] added a mutation operator to the PSO algorithm. In an iteration, the algorithm determines whether the current local and the global best position change. If yes, the normal PSO will continue and if no, the mutation mechanism of the GA interferes with the particle, and this might prevent the particle getting stuck in the local minimum [57].

In the proposed hybrid algorithm, the initial population is randomly generated, and then the crossover and mutation operators are applied in an iteration. In order to develop generation strategies, two types of new operators inspired by PSO algorithm (called PSO operators 1 and 2) are proposed. At a rough guess, three types of new generations are found in the proposed methodology:

- Children generated by crossover operations via all current populations.
- Children produced by offspring of the best ever solution (gbest) with all current populations (called PSO operator 1), and
- Children generated by offspring of the best ever result of a single solution (pbest) with all current populations (called PSO operator 2).

Finally, the mutation operator is interfered through a limited random rate after every offspring. All the steps of the proposed algorithm are presented as follows (an example is given with each algorithm explanation step to ensure a better view of the algorithm):

1. Make an initial random population (chromosomes/particles) as basic starting solutions (for example take 100 initial random solutions) to generate an initial population.
2. Calculate the fitness for all solutions according to the model's objective function.
3. Update pbest for any single solution and gbest (from iteration number two to the end).
4. Use a RAR-type crossover operator [40], to generate children via the current population (all parents). This operator produces one child from each couple (makes 50 children in the given example from 100 parents).
5. Use PSO operator 1: each member of the current population (particles) makes another crossover process with its pbest (100 more children).
6. Use PSO operator 2: each member of the current population (particles) makes the final crossover with current gbest (100 more children).
7. Undertake mutation operator by a limited predetermined random rate.
8. Calculate fitness for all solutions.
9. Choose the best population-size solutions (100 in the presented example) between all new children by the GA operator and both PSO operators and all parents to shape the next generation (the best of the first 100 solutions will be the next generation).
10. REPEAT from line 3.

Fig. 3 shows the total procedure for the developed hybrid algorithm. Complete explanations of Fig. 3 are prepared to discuss the steps of the proposed algorithm.

It is essential to describe all steps in detail to clarify the proposed algorithm:

Step 1: A difficulty in such algorithms for large problems is producing feasible solutions. It is very difficult to generate the values of flows (allocation variables) randomly. On the other hand, binary decision variables (location) are utilized for some critical steps of the proposed algorithm like steps 1, 2, 5, 6, and 7. This ensures that the algorithm achieves reasonable results. To begin with the configuration network is first set up (activated suppliers, manufacturers, distributors, etc.) through binary variables, and then the initial configuration by various operators of the proposed algorithm is evolved. A simple heuristic algorithm is built to handle network flows. The most important issue is the capacity constraint, which results in infeasible solutions. Hence, an estimation about flows of network (flows from suppliers to manufacturers in each period) based on demands of customers per period and capacities of activated facilities is made. Finally, rules are made to generate the feasible network values:

- Only activated facilities are part of flows.
- Facilities with higher capacities get higher values. This rule is utilized in a normalized matrix of capacities, which provides a reasonable matrix to assign a value to some facilities.

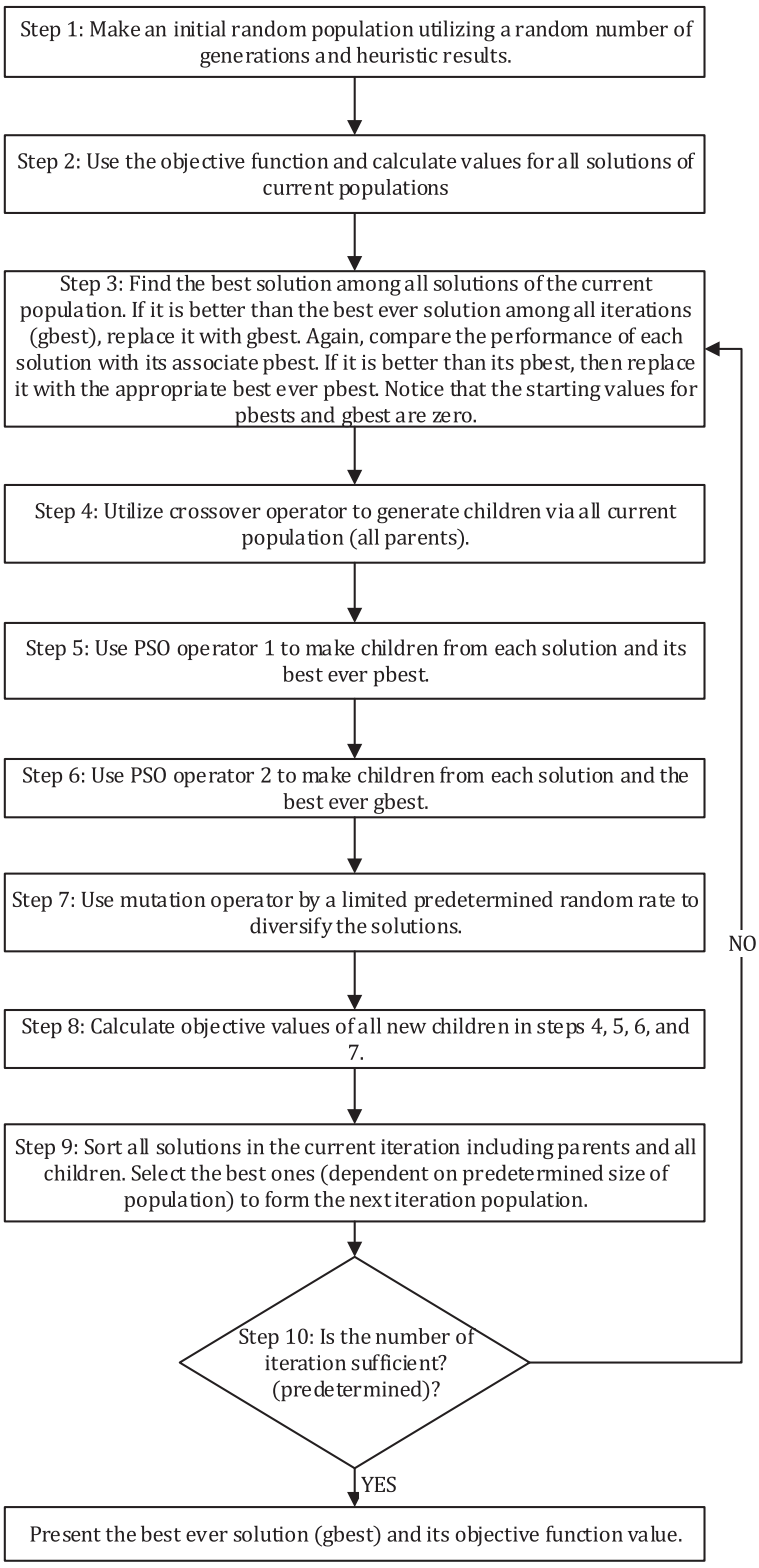

Fig. 3. Total steps of the proposed PSO-GA hybrid methodology.

- If some facilities (for examples manufacturers) have flows with different types of facilities (manufacturers to distributers and to warehouses), reasonable rates should generate constructing balanced flows.

At the end of step 1, some basic feasible solutions are generated as initial population.

Step 2: The objective function maximizes the profit and also calculates the desirability of each solution of. It is a utility criterion of solutions. The end of this step leads to an acceptability measure of each solution in the population.

Step 3: In iteration number 2 or more, the values of "pbests" and "gbest" should be compared with current values of corresponding "pbests" and "gbest" to preserve the better ones. At the conclusion of this step, "pbests" and "gbest" are updated.

Step 4: The RAR operator is a one-point crossover, which is performed on location decision variables. The rules are also simple: If both parents activate a special facility, then that facility is activated in their child. And vice versa: if both parents do not have a special facility, then that facility is not activated in their child. Otherwise, a final zero-one value will be randomly assigned. At the end of this step, we have one new solution (child) through two parental solutions.

Step 5: Here, the PSO operator called pbest is utilized. Each current solution in iteration has a background improvement process. A solution's best performance in its background is called pbest (local best solution). All pbests are preserved in the proposed algorithm (inspired by the PSO algorithm). In this step, each solution engenders a corresponding pbest and creates a new child through a one-point crossover as explained in step 4. This step ends with a new child from each solution.

Step 6: The best ever solution in all iterations is gbest (global best solution). Inspired by the PSO algorithm, the proposed algorithm preserves gbest. In this step, each solution engenders a current gbest and creates a new child by one point crossover as explained in step 4. The end of this step leads to a new child from each solution.

Step 7: Mutation operation is undertaken on location decision variables with a limited rate. Actually, this operator is implemented to some portions of the population. It can activate/inactivate a facility. This operator can elevate the diversification capabilities of the proposed algorithm.

Step 8: Utilizing objective function, the profits of all new children are calculated. As explained above, these children are generated by the normal crossover operator (step 4) and PSO operators (steps 5 and 6).

Step 9: All new solutions (children) including current solutions (current population) are sorted out and the best ones selected to form the next iteration population. As the size of the population is known, so the numbers of best-selected solutions are also known.

Step 10: The above steps should be repeated until the goal of achieving the total number of iterations is reached.

A simple tuning procedure is performed utilizing a mid-size instance to find the best parameter setting for the crossover rate, mutation rate, population size, and sufficient iteration numbers of the methodologies. Such analyses determine the appropriate characteristics of the GA and hybrid algorithm to ensure the reliability of results. In order to set the parameters of both, focus is on mutation rate, crossover rate, population size, and iteration number, which are determined with proper

Table 3
Computational study parameters.

Row	Parameter	Uniform distribution or rate
1.	Demand	0–3000
2.	Second demand rate	50% of demand
3.	Price	3750–5000
4.	Second product price	50% of price
5.	Purchasing cost	10% of price
6.	Manufacturer capacity	6000–14,000
7.	Remanufacturer capacity	50% of manufacturer capacity
8.	Supplier capacity	18,000–42,000
9.	Supplier recycling capacity	50% of supplier capacity
10.	Recycling cost	10–100
11.	All other reverse cost	10–100
12.	Other facility capacities	6000–14,000
13.	Material cost	100–1000
14.	Manufacturing cost	100–1000
15.	All other forward cost	100–1000
16.	Shortage cost	1000–5000
17.	Supplier fixed cost	7–10 million
18.	Manufacturer fixed cost	70–150 million
19.	Distributor fixed cost	1–2 million
20.	Warehouse fixed cost	0.1–1 million
21.	Disassembly fixed cost	0.1–1 million
22.	Redistributor fixed cost	0.1–1 million
23.	Disposal center fixed cost	0.1–1 million
24.	Batch size	1

instances. First, based on Table 3, a mid-size instance is considered and then the problem is run many times. Finally, the best values are achieved by analyzing the performances of the algorithms in different parameter settings.

The validation phase is the most important part of proposing an algorithm. A complete validation process based on the model is available. The procedure has two steps; small-sized and large-sized instances, described in the next section.

5. Validation process of the proposed hybrid algorithm

In order to evaluate the new hybrid algorithm, two validation processes-small size and large-scale instances – are arranged. First, a small-sized instance is developed due to its ability to compare with the global optimum. This model is coded by IBM ILOG CPLEX 12.2 optimization software to achieve the global optimum. Second, the proposed hybrid algorithm is coded in a very complicated process utilizing Matlab 7.12 (R2011a) software. Finally, a GA is coded by Matlab software to complete the process of comparison. All computations are run on a Core 2 Duo-2.26 GHz processor laptop. Parameters are generated using uniformly distributed data as presented in Table 3. It should be noted that the recycling, remanufacturing, repair, and disposal rates are 0.2, 0.4, 0.3, and 0.1, respectively. In addition, batch sizes are considered as one and the mutation rate is 0.2.

The detail of computational analysis is presented as follows.

5.1. Small-sized instances analysis

In order to evaluate the effectiveness of the proposed algorithm, a small and acceptable-sized instance that contains five units of each entity is generated. Thus, there will be five suppliers, five manufacturers, five warehouses, etc., for five products in five periods, where there are 14,482 constraints and 11,336 decision variables including 460 binary variables. In this phase of validation, 10 different kinds of instances are generated and the objective function of CPLEX (global optimum), proposed hybrid, and genetic algorithm are recorded. The population size is 100 for the proposed algorithm, the GA, and PSO. The iteration number is set at 120. These values are set based on a simple tuning procedure. All algorithms achieve their best solutions within a reasonable time (less than 3 min for all algorithms). Results are illustrated in Table 4 to analyze objective function values and in Fig. 4 to analyze the duration of achieving optimal points.

Analyzing the results of Tables 4 and 5 and Fig. 4, the following points can be concluded:

- The differences between the global optimum of the CPLEX software and the proposed hybrid algorithm are sufficiently small. When the results are reviewed, just 10% difference from the global optimum is admissible to persuade the acceptability of the proposed algorithm's performance. It should be mentioned that the GA provides a normal performance with 20% average distance to the global optimum. Further, PSO, with 18% difference to the global optimum, provides better performances in comparison with the GA. However, both the GA and PSO could not achieve solutions near the hybrid algorithm. These results can prove high capabilities of the proposed hybrid algorithms in achieving better solutions.
- The performance of the proposed hybrid algorithm is 10% and 8% better than the similar GA and PSO, respectively. Also in all instances without any exceptions, the proposed hybrid algorithm presents better performance than the GA (see Fig. 4) and PSO. As discussed before, this was predictable based on the new evolutionary characteristics of the proposed algorithm.
- Time analysis in Table 5 proves the acceptable behaviors of the GA, PSO, and the proposed hybrid algorithm with the average of 95, 98, and 139 s to achieve the optimal points, respectively.

Finally, total results illustrate that the proposed hybrid algorithm increases the potential of achieving more-robust solutions and can make acceptable synergy between GA and PSO. Indeed, in terms of the quality of solutions, the proposed hybrid algorithm provides a significantly better solution and it can be the most important advantage of this algorithm. On the other

Table 4
Results of objective functions for small-sized instances (in millions).

Instance	CPLEX global optimum	Proposed hybrid algorithm	Differences to optimum (%)	GA	Differences to optimum (%)	PSO	Differences to optimum (%)
1	463	442.4	5	397.5	14	386.3	17
2	444	389.9	12	363.9	18	340.6	23
3	431	394.3	8	325.5	24	346.8	19
4	446	403.7	10	367.9	18	395.8	11
5	595	512.3	14	425.3	29	452.9	24
6	574	502.1	13	447.8	22	456.6	20
7	559	499.6	11	428.3	23	460.7	18
8	563	489.3	13	422.2	25	455.9	19
9	323	286.8	11	277.7	14	277.0	14
10	850	781.7	8	712.2	16	742.3	13
Average	525	468	10	417	20	432	18

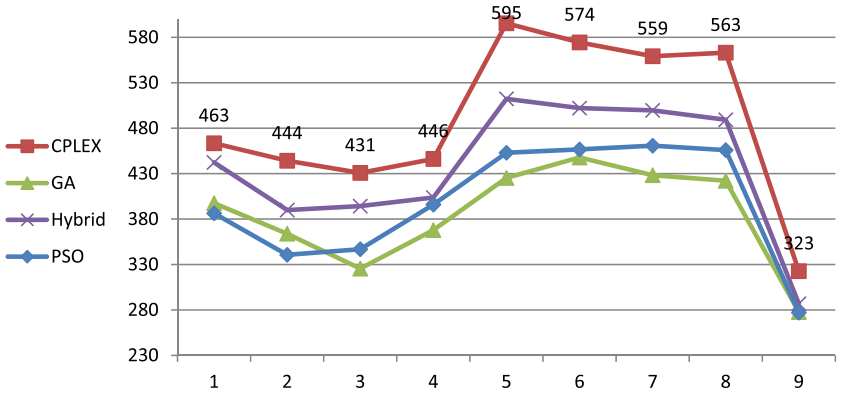

Fig. 4. Results of small-sized instances (in millions).

Table 5
Time analysis for small-sized instances (in seconds).

Instance	CPLEX	Hybrid algorithm	GA	PSO
1	80	130	90	95
2	105	140	100	92
3	108	135	105	110
4	100	142	100	106
5	95	151	103	123
6	106	136	108	95
7	90	145	95	100
8	94	149	96	98
9	50	132	83	87
10	45	129	69	79
Average	87	139	95	98

hand, in terms of time, the proposed algorithm could not present better performances and it could be the major disadvantage of the proposed algorithm. However, when the size of instances become larger, general exact solvers cannot achieve the optimum (or even reasonable) solutions, and utilizing reliable metaheuristic algorithms (such as the developed hybrid algorithm) is unavoidable. Besides, if we have enough time, the algorithm should have the capabilities of achieving better solutions. In this aspect, the proposed algorithm illustrates acceptable capabilities.

Consequently, reviewing the acceptable performance of the proposed hybrid algorithm, it passes validation procedure when compared to the CPLEX global optimum. Consequently, these two metaheuristic algorithms in large and real scale instances can be evaluated for our problem.

5.2. Further evaluations

The objective of this section is validation of the proposed hybrid algorithm in large-scale instances. In the previous section, the acceptability of the proposed algorithm results was guaranteed. A total of 13 different large sizes of instances was prepared and presented in Table 6 for computational study. The characteristics of all instances are clarified in mentioned table (see Table 6). All evaluations are performed in 12 periods for 100 iterations.

Reviewing Table 6 reveals larger problems than those seen in earlier researches. The proposed algorithm is compared with two types of GA; the first has the same population size as the hybrid algorithm and the second has double the size to ensure hybrid's prominence. The results are illustrated in Tables 7 and 8 and Fig. 5. It should be noted that CPLEX could not achieve any reasonable solutions even after a day of running any of the instances in Table 7.

By analyzing of the results of Tables 7 and 8 and Fig. 5, the following conclusions are reached:

- The proposed hybrid algorithm shows significantly better performance than the genetic algorithm. The results of Table 7 reveal a total of 28% better objective function values of the proposed algorithm. Even when the population size of the GA is duplicated, 18% better results are achieved. In some cases like instances 4 and 11, the doubled-sized GA shows a worse performance than the default GA, thereby proving that the GA cannot guarantee better performances by duplicating population size.
- Based on the schematic view of Fig. 5, the proposed hybrid algorithm reveals better performance than two GA types in all 13 instances without exception. Hence, its prominence can be judged. Further, more distinctions in objective functions value in larger instances can be observed. It may be concluded that in the larger the instances, more reliable results can be expected from the proposed algorithm.

Table 6
Characteristics of all instances.

Instance number	Number of					
	Hybrid population	GA population	Suppliers, warehouses, distributors, redistributors	Manufacturers, disposal, disassembly centers	Customers, second customers	Products
1.	20	20 and 40	60	15	60	6
2.	20	20 and 40	65	16	65	7
3.	20	20 and 40	70	17	70	7
4.	20	20 and 40	75	18	75	7
5.	20	20 and 40	80	19	80	8
6.	20	20 and 40	85	20	85	8
7.	20	20 and 40	90	21	90	9
8.	20	20 and 40	95	22	95	9
9.	20	20 and 40	75	18	75	10
10.	20	20 and 40	100	25	100	10
11.	20	20 and 40	150	30	150	15
12.	20	20 and 40	190	45	200	20
13.	20	20 and 40	200	50	200	20

Table 7
Results of large-scale analysis.

Instance	Profit of different methodologies (in millions)					
	Hybrid algorithm	Genetic algorithm	Difference between hybrid and GA (%)	GA double size population	Difference between hybrid and GA doubled (%)	Number of decision variables
1.	17,921	11,399	36	15,409	14	32,235
2.	22,467	19,039	15	21,472	4	37,610
3.	24,036	20,000	17	20,359	15	43,399
4.	25,528	21,281	17	19,005	26	49,602
5.	31,854	14,657	54	25,416	20	56,219
6.	33,341	12,877	61	23,761	29	63,250
7.	39,013	28,984	26	33,650	14	70,695
8.	39,675	32,344	18	36,662	8	78,554
9.	38,290	27,129	29	31,932	17	49,602
10.	53,909	50,258	7	51,935	4	89,225
11.	57,081	42,042	26	30,842	46	186,990
12.	205,400	131,850	36	176,380	14	355,950
13.	215,210	172,360	20	174,110	19	355,950
Average	61,825	44,940	28	50,841	18	113,022

Table 8
Results of large-scale time analysis of methodologies.

Instance	Durations of different methodologies (in minutes)		
	Hybrid algorithm	Genetic algorithm	Doubled-sized population GA
1.	17.8	7.0	13.4
2.	23.5	8.65	17.4
3.	18.6	6.7	13.9
4.	20.3	8.05	15.3
5.	25.2	10.3	19.3
6.	28.0	11.6	22.3
7.	35.1	15.3	29.7
8.	38.8	15.5	30.1
9.	25.3	11.4	21.9
10.	45.1	17.9	39.2
11.	42.0	17.7	34.0
12.	115.0	47.0	80.0
13.	120.0	48.0	88.0
Average	42.7	17.3	32.7

- Analyzing decision variables clarify the large sizes of the instances. The necessity of an efficient algorithm, giving acceptable results at proper times is clarified when tremendous problems exist. In instances 12 and 13, there are around 356,000 of decision variables including 950 binary variables. Clearly, just assigning the 950 binary variables leads to 95×10^{285} different combinations. Consequently, the necessity for an acceptable and well-performing algorithm for such problems is revealed.

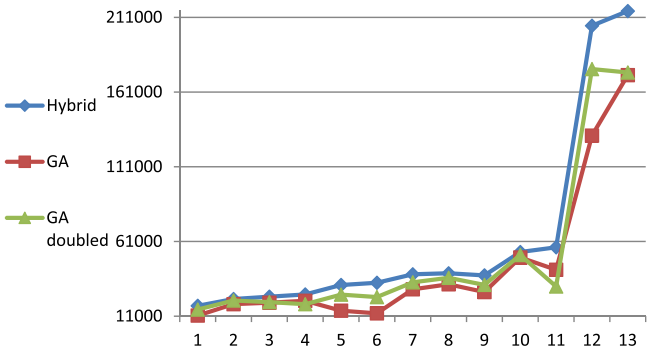

Fig. 5. Results of large-scale analysis (in millions).

- The average time for achieving results, illustrated in Table 8, proves the acceptable performance of the proposed hybrid algorithm, which achieves optimal points after processing all iterations in around 43 min, which is admissible in such tremendous instances. Though the GA performs better than the proposed hybrid in terms of time, better quality solutions can cover the differences between these two.

The proposed algorithm's last performance evaluation point in this section is about its performance when compared with the PSO. It can be expected that the proposed algorithm presents better results in comparison with PSO according to the short analyses in Table 2. As discussed, the proposed algorithm strives for synergy between GA and PSO. Indeed, it attempts an elevated algorithm by utilizing the positive characteristics of GA and PSO simultaneously. Therefore, it might provide better performance when compared with the PSO, too.

As analyzed, the performance of this new proposed hybrid algorithm is better than the GA. Such results are predictable as the GA was evolved with the strengths of particle swarm optimization, which can also be argued regarding the preponderance of the proposed algorithm on PSO.

6. Sensitivity analysis

As the problem of this paper is a location–allocation, and as fixed costs play important roles, the sensitivity of the solution methodologies in this criterion is evaluated. Hence, the special instance where five units are considered in each entity is regarded (the worst case of Table 5) to find the sensitivity. Performances of the proposed hybrid algorithm, GA, and PSO are compared with the optimum points of CPLEX in five cases of fixed costs based on Table 3 parameters: decreasing 50% in fixed costs, decreasing 25% in fixed costs, the normal fixed costs of Table 3, increasing 50% in fixed costs, and increasing 25% in fixed costs. For instance, in the first row of Table 9, the phrase "−50%" means decreasing opening costs of all entities to 50% of their original values to solve the problem. Finally, objective function values are reported. It is expected that methodologies will behave similarly to the original values (row "0%") as regards differences with optimum points. The results are illustrated in Table 9 and Fig. 6. The results are achieved after sufficient runs.

Under considerations of sensitivity analysis, as illustrated in Table 9, and Fig. 6, all evaluated algorithms (GA, PSO, and the proposed hybrid) present acceptable behaviors in different fixed costs of all entities. In the original instance, determined in the third rows of Table 9 (case of 0%), the 14%, 29%, and 24% differences to the optimum points can be observed for the hybrid, GA, and PSO, respectively. These differences somehow remain in other sensitivity cases. The average differences for these methodologies are 14% (for the hybrid), 25% (for the GA), and 23% (for PSO), which can prove acceptable performances of all developed methodologies in different cases of opening costs.

The proposed hybrid algorithm could preserve its average difference in comparison with the original value (14% on average in comparison with original value of 14%). Fig. 6 illustrates the trends of three methodologies compared to the trend of CPLEX in changing fixed costs. Roughly speaking, all algorithms present acceptable trends in the fixed costs sensitivity analysis but the proposed hybrid algorithm shows better behavior when compared to GA and PSO.

7. Managerial implications and case study in a hospital furniture manufacturer

This research attempts to develop a new solution methodology in CLSC network design. Therefore, by optimizing the network configurations, managerial capabilities in elevating their CLSC and also solving real-world problems are improving. Such improvements can help practitioners in applying theoretical developments for real cases.

On the other side, when managers decide to apply such models in their organization, problems result in most cases, which general exact solvers are unable to cope with. In addition, the results of the model have tremendous effects on management decisions for location and allocation of the CLSC network. Hence, elevating the capabilities of solving such vital problems and proposing new methodologies which can solve real cases in appropriate time with reliable solutions, are important steps to

Table 9
Results of sensitivity analysis of fixed costs in objective function values (in millions).

Fixed costs changes	CPLEX global optimum	Proposed hybrid algorithm	Differences to optimum (%)	Genetic algorithm	Differences to optimum (%)	Genetic algorithm	Differences to optimum (%)
−50%	662	613	7	534	19	523	21
−25%	628	524	17	475	24	501	20
0%	**595**	**512**	**14**	**425**	**29**	**453**	**24**
+25%	566	474	16	406	28	428	24
+50%	537	453	16	399	26	403	25
Average	**598**	**511**	**14**	**448**	**25**	**462**	**23**

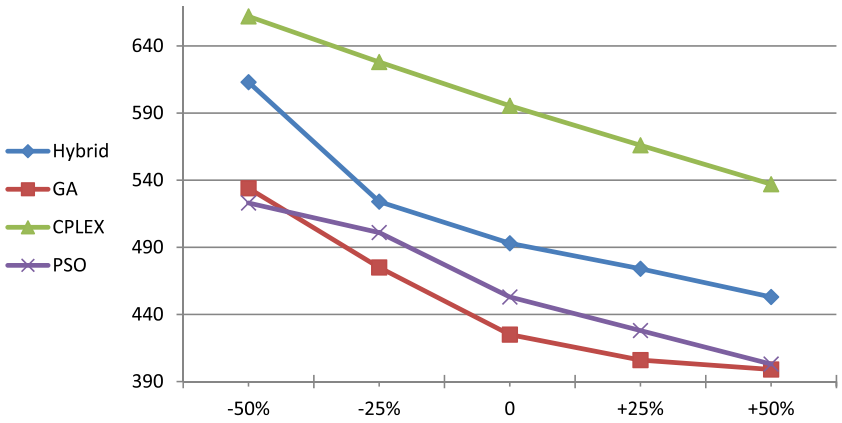

Fig. 6. Results of sensitivity analysis of fixed costs in objective function values (in millions).

Table 10
Parameters of case study (amounts are in Rials of Iran*).

Row	Parameter	Beds	Stretchers
1.	Demands of the first customers per month (per period)	167	84
2.	Amount of return products per month	34	17
3.	Price of first products	9,000,000	6,500,000
4.	Price of second products	6,700,000	4,800,000
5.	Purchasing price of used products from customers	420,000	300,000
6.	Number of main suppliers	6	7
7.	Number of distributors and redistributors	0: It is done by third party providers and paid by customers	
8.	Manufacturing capacity per month	200	100
9.	Warehouse capacity per month	20	10
10.	Remanufacturing capacity per month	108	36
1.1.	Repairing capacity per month (if it exists in the health system)	70	20
12.	Recycling capacity per month (actually, it is disassembling to prepare some raw materials)	35	15
13.	Manufacturing costs per product	2,500,000	1,600,000
14.	Remanufacturing costs per product	2,000,000	1,280,000
15.	Repairing costs per product	250,000	160,000
16.	Recycling costs per product	200,000	150,000
17.	Disposal costs per product	1,960,000	760,000
18.	Supplying costs per product	2,940,000	3,724,000
19.	Holding costs per product per month	400000	100000
20.	Distributing costs per product	0 (done by customers)	
21.	Shortage costs per product per month	270,000	195,000
22.	Repairing rate	0 (rare in health systems)	
23.	Remanufacturing rate	55%	65%
24.	Recycling rate	25%	15%
25.	Disposal rate	20%	20%
26.	Percentage of raw materials added by return products (from recycling)	4%	3%

**Note:* One Euro is equal to 32,000 Rials of Iran.

applicability of CLSC problems in real situations for managers. Therefore, the results of this paper present powerful and reliable methods for managers to cope with CLSC network design and planning issues. It also encourages managers to revise the location–allocation decisions of the CLSC network of their organizations.

Table 11
Results of the case study analysis (in millions of Rials).

Criteria	CPLEX global optimum	Proposed hybrid algorithm	Differences to optimum (%)	Genetic algorithm	Differences to optimum (%)
Total profit	7795.668	7369.603	5	7295.907	6
Total first sales	24588	23436	5	21092.4	14
Total second sales	181.62	177.1762	2	168.3174	7
Purchasing costs	189.6	172.489	9	162.1396	14
Recycling costs	18.7	17.27309	8	16.58216	11
Remanufacturing costs	526.56	489.888	7	484.9891	8
Average	–	–	6	–	10

Actually, in developing countries, there is a growing interest among companies in dealing with the forward and reverse supply chains simultaneously. Also, customers are aware of not disposing of used products directly. Hence, these studies can help managers to close the loop of their supply chain while achieving more profit instead of just thinking of cost-decreasing methods in closing the loop. To shed more light on the results of this paper, a case study of a hospital furniture company in Iran is investigated and the results are analyzed from the managerial points of view.

"Mehran Hospital Industries" (Mehran Teb Med Co., www.Mehranmed.com) is the first (since 1952) and the largest manufacturer of hospital furniture in Iran producing different kinds of beds, stretchers, trolleys, and cabinets for hospitals. Recently, this company started to invest in closing its supply chain loop by collecting used products from hospitals and dealing with them in different ways. Since, hospital furniture plays a very important role in the health system, they extended their reverse supply chain precisely. In order to apply the model and methodology of this paper, Table 10 is provided to determine the parameters of the model by holding some meetings with the general manager of the company and also other managers.

Based on the data of Table 10, optimum flows in the closed-loop supply chain network of the company are analyzed. It should be mentioned that the two main products of the company (beds and stretchers) are the only products in the reverse loop of the company yet. The company has three manufacturing sites that include some cells to provide reverse procedures. Disposing activities are outsourced to appropriate centers. The problem is solved for two products (main products) in 12 periods (12 months) and the results of the three algorithms are presented in Table 11.

Reviewing Table 11 clarifies some valuable points of the real case study and the performance of the proposed hybrid algorithm. Several criteria of the results to solve the case study problem are illustrated in Table 11, which evaluates the performances of the GA and the proposed hybrid algorithm in detail. Total profit, total first sales, total second sales, and the main reverse costs like purchasing costs, recycling costs, and remanufacturing costs are mentioned. In all criteria, the differences between the global optimum of CPLEX and the results of the GA and the hybrid algorithm are calculated and reported. The proposed hybrid algorithm presents an acceptable behavior to provide near-to-optimal results with a total average of 6% differences and 5% distance to optimal value. It performs better than the GA in terms of quality of solutions. An interesting point for the company is the ability of the proposed solutions to make the reverse loop feasible and profitable. The main recommendation to the managers of the company would be to extend the capacity of the reverse supply chain to ensure better feasibility and profits. Another recommendation is providing some motivations for the customers to encourage them to sell/replace their used products via manufacturer. This will increase the current rate of return of products (20%) and make the reverse loop more feasible.

Finally, it can be concluded that the proposed hybrid algorithm provides an acceptable performance in real cases of a hospital furniture leader company that can confirm its applicability in real-world situations. Indeed, in comparison with the GA and PSO, the developed hybrid algorithm can achieve more-robust and more-reliable solutions in real cases. In this case study, small instances are considered in order to observe the global optimum. However, in larger instances, CPLEX is completely unable to present solutions and therefore, we have to consider other reliable approaches such as the proposed metaheuristics.

8. Conclusion and future research

This paper copes with an important practical issue regarding design and planning of a closed-loop supply chain in real and advanced systems. Despite some claims, few papers consider large-scale problems in the field of CLSC. Therefore, first a comprehensive model is presented (multi-echelon, multi-product, multi-period, and appropriate generalizations in network flows). Second, a new PSO–GA hybrid algorithm is proposed to solve various kinds of problems. A complete computational analysis is undertaken to validate the proposed algorithm. First, the model is coded by IBM ILOG CPLEX 12.2 optimization software to achieve global optimum. Meanwhile, the proposed algorithm and the GA are coded using Matlab 7.12 (R2011a) software. Second, 10 small-sized instances are developed and the performances of the proposed algorithm and the GA are compared to the global optimum of CPLEX. The results show 6% and 15% distance to the global optimum for the proposed hybrid algorithm and the genetic algorithm, respectively. Third, 13 different sizes of instances including very

large-scale ones (with 356,000 decision variables) are generated. The results of the proposed algorithm are compared with the performances of two types of GA. The results show that the proposed algorithm has 28% better objective functions and depends on the size of the problems with these distinctions increasing. Managerial insights and a case study on a large Iranian hospital furniture manufacturer are investigated in detail with the achievements of the proposed hybrid algorithm being noticeable. Roughly speaking, the performance of the new proposed hybrid algorithm is an improvement over genetic algorithms in all instances without exceptions.

This also reveals some guidelines for future research; first, other approaches such as simulation-based optimization can be applied instead of a mathematical programming approach, which can lead to interesting findings. Second, the single objective approach can be elevated to multi-objective approaches, which let the network consider some essential objectives such as minimizing CO_2 emissions and shortage of products and maximizing the number of customer respondents. Third, the grading of products can be regarded in the reverse logistics of the network so that different purchasing prices, reprocessing costs and strategies, and selling prices can be considered for each grade. Fourth, the proposed algorithms could be compared with other metaheuristics like tabu search, ant colony, and simulated annealing. At last, the deterministic approach of this paper could be improved by considering non-deterministic parameters, especially uncertainties in time, quality, and quantities of return products. If we cannot count on return products characteristics, closing the loop of the supply chain becomes completely infeasible. Hence, this is a critical direction for future research.

Acknowledgments

The authors would like to thank editor-in-chief Professor Johann (Hans) Sienz and the referees for their comments that significantly improved the earlier version of this paper. We are also grateful to Mr. Farzin Akbari Nakhjavani, general manager of Mehran Hospital Industries (Mehran Teb Med Co.) for his support in providing real data of the company and accessing different documents of the closed-loop supply chain of the company.

Appendix A. The model formulation

The complete proposed 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 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 product, indexed by "u".
T: number of period, indexed by "t".

Parameters

D_{cut}: demand of product "u" by the first customer "c" in period "t".

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 cost 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 cost 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 of manufacturer "f" of product "u" in period "t".
RFC_{fut}: remanufacturing capacity of manufacturer "f" of product "u" in period "t".
WC_{wut}: warehouse capacity in hours of warehouse "w" of product "u" in period "t".
DC_{dut}: capacity of distributor "d" of product "u" in period "t".
AC_{aut}: capacity of disassembly "a" of product "u" in period "t".
RC_{rut}: capacity of redistributor "r" of product "u" in period "t".
PC_{put}: capacity of disposal center "p" of product "u" in period "t".

Mc_{sut}: material cost of product "u" per unit supplied by supplier "s" in period "t".
Rc_{sut}: recycling cost of product "u" per unit recycled by supplier "s" in period "t".
Fc_{fut}: manufacturing cost of product "u" per unit manufactured by manufacturer "f" in period "t".
RFc_{fut}: remanufacturing cost of product "u" per unit by manufacturer "f" in period "t".
DAc_{aut}: disassembly cost of product "u" per unit by disassembly center "a" in period "t".
RPc_{aut}: repairing cost of product "u" per unit repaired by disassembly location "a" in period "t".
Pc_{aut}: disposal cost of product "u" per unit disposed by disposal location "p" in period "t".
Nc_{fut}: non-utilized manufacturing capacity cost of product "u" of manufacturer "f" in period "t".
RNc_{fut}: non-utilized remanufacturing cost of product "u" of manufacturer "f" in period "t".
Sc_{ut}: shortage cost of product "u" per unit in period "t".
Fh_{fu}: manufacturing time of product "u" per unit at manufacturer "f".
RFh_{fu}: remanufacturing time of product "u" per unit at manufacturer "f".
Rc_{sut}: recycling cost of supplier "s" of product "u" in period "t".
WH_{wut}: holding cost of product "u" per unit at the warehouse "w" in period "t".
DH_{dut}: holding cost of product "u" per unit at distributor store "d" store 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", manufacturer "f", distributor "d", disassembly "a", redistributor "r", warehouse "w" and customer "c" respectively.
Tc_{ut}: transportation cost 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.
M: is a large number.

Decision variables

L_i: binary variable equals "1" if location "i" is open and "0" 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",
R_{dut}: the residual inventory of product "u" at distributor "d" in period "t".

A.1 Objective function

We have considered profit as the objective function, so we should calculate all sales and costs:
Total sales:
Sales of all products:
First products sale (flows from distributors, manufacturers, and warehouses):

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

Second products sale (flows from redistributors, manufacturers, and warehouses):

$$\sum_{r\in R}\sum_{k\in K}\sum_{u\in U}\sum_{t\in T}Q_{rkut}B_{ru}P_{kut}+\sum_{f\in F}\sum_{k\in K}\sum_{u\in U}\sum_{t\in T}Q_{fkut}B_{fu}P_{kut}+\sum_{w\in W}\sum_{k\in K}\sum_{u\in U}\sum_{t\in T}Q_{wkut}B_{wu}P_{kut}. \tag{2}$$

Total costs:
Total costs = fixed costs + material costs + manufacturing costs + non-utilized capacity costs + shortage costs + purchasing costs + disassembly costs + recycling costs + remanufacturing costs + repairing costs + disposal costs + transportation costs + inventory holding costs.
Fixed costs (location costs):

$$\sum_{s\in S}F_sL_s+\sum_{f\in F}F_fL_f+\sum_{d\in D}F_dL_d+\sum_{a\in A}F_aL_a+\sum_{r\in R}F_rL_r+\sum_{p\in P}F_pL_p+\sum_{w\in W}F_wL_w. \tag{3}$$

Material costs:

$$\sum_{s\in S}\sum_{f\in F}\sum_{u\in U}\sum_{t\in T}Q_{sfut}B_{su}Mc_{sut}-\sum_{a\in A}\sum_{s\in S}\sum_{u\in U}\sum_{t\in T}Q_{asut}B_{au}(Mc_{sut}-Rc_{sut}). \tag{4}$$

Manufacturing costs:

$$\sum_{f\in F}\sum_{d\in D}\sum_{u\in U}\sum_{t\in T}Q_{fdut}B_{fu}Fc_{fut}+\sum_{f\in F}\sum_{w\in W}\sum_{u\in U}\sum_{t\in T}Q_{fwut}B_{fu}Fc_{fut}+\sum_{f\in F}\sum_{c\in C}\sum_{u\in U}\sum_{t\in T}Q_{fcut}B_{fu}Fc_{fut}+\sum_{f\in F}\sum_{k\in K}\sum_{u\in U}\sum_{t\in T}Q_{fkut}B_{fu}Fc_{fut}. \tag{5}$$

Non-utilized capacity costs (for manufacturers):

$$\sum_{f\in F}\left(\sum_{u\in U}\left(\sum_{t\in T}\left((FC_{fut}/Fh_{fu})L_f-\sum_{d\in D}(Q_{fdut}B_{fu})-\sum_{w\in W}(Q_{fwut}B_{fu})-\sum_{c\in C}(Q_{fcut}B_{fu})+\sum_{w\in W}\sum_{r\in R}Q_{wrut}B_{wu}+\sum_{w\in W}\sum_{k\in K}Q_{wkut}B_{wu}\right)Nc_{fut}\right)\right)$$
$$+\sum_{f\in F}\left(\sum_{u\in U}\left(\sum_{t\in T}\left((RFC_{fut}/RFh_{fu})L_f-\sum_{r\in R}(Q_{frut}B_{fu})-\sum_{k\in K}(Q_{fkut}B_{fu})-\sum_{w\in W}\sum_{r\in R}Q_{wrut}B_{wu}+\sum_{w\in W}\sum_{k\in K}Q_{wkut}B_{wu}\right)RNc_{fut}\right)\right). \tag{6}$$

Shortage costs (for distributor):

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

Purchasing costs:

$$\sum_{c\in C}\sum_{a\in A}\sum_{u\in U}\sum_{t\in T}Q_{caut}PH_{cut}B_{cu}. \tag{8}$$

Disassembly costs:

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

Recycling costs:

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

Remanufacturing costs:

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

Repairing costs:

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

Disposal costs:

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

Transportation costs:

$$\sum_{t\in T}\sum_{u\in U}\sum_{s\in S}\sum_{f\in F}Q_{sfut}B_{su}Tc_{ut}DS_{sf}+\sum_{t\in T}\sum_{u\in U}\sum_{f\in F}\sum_{d\in D}Q_{fdut}B_{fu}Tc_{ut}DS_{fd}+\sum_{t\in T}\sum_{u\in U}\sum_{f\in F}\sum_{w\in W}Q_{fwut}B_{fu}Tc_{ut}DS_{fw},$$

$$\sum_{t\in T}\sum_{u\in U}\sum_{f\in F}\sum_{c\in C}Q_{fcut}B_{fu}Tc_{ut}DS_{fc}+\sum_{t\in T}\sum_{u\in U}\sum_{f\in F}\sum_{k\in K}Q_{fkut}B_{fu}Tc_{ut}DS_{fk}+\sum_{t\in T}\sum_{u\in U}\sum_{w\in W}\sum_{c\in C}Q_{wcut}B_{wu}Tc_{ut}DS_{wc},$$

$$\sum_{t\in T}\sum_{u\in U}\sum_{w\in W}\sum_{k\in K}Q_{wkut}B_{wu}Tc_{ut}DS_{wk}+\sum_{t\in T}\sum_{u\in U}\sum_{d\in D}\sum_{c\in C}Q_{dcut}B_{du}Tc_{ut}DS_{dc}+\sum_{t\in T}\sum_{u\in U}\sum_{a\in A}\sum_{s\in S}Q_{asut}B_{au}Tc_{ut}DS_{as},$$

$$\sum_{t\in T}\sum_{a\in A}\sum_{u\in U}\sum_{f\in F}Q_{afut}B_{au}Tc_{ut}DS_{af}+\sum_{t\in T}\sum_{u\in U}\sum_{a\in A}\sum_{p\in P}Q_{aput}B_{au}Tc_{ut}DS_{ap}+\sum_{t\in T}\sum_{u\in U}\sum_{a\in A}\sum_{r\in R}Q_{arut}B_{au}Tc_{ut}DS_{ar},$$

$$\sum_{t\in T}\sum_{u\in U}\sum_{f\in F}\sum_{r\in R}Q_{frut}B_{fu}Tc_{ut}DS_{fr}+\sum_{t\in T}\sum_{u\in U}\sum_{w\in W}\sum_{r\in R}Q_{wrut}B_{wu}Tc_{ut}DS_{wr}+\sum_{t\in T}\sum_{u\in U}\sum_{r\in R}\sum_{k\in K}Q_{rkut}B_{ru}Tc_{ut}DS_{ruk},$$

$$\sum_{t\in T}\sum_{u\in U}\sum_{c\in C}\sum_{a\in A}Q_{caut}B_{cu}Tc_{ut}DS_{ca}+\sum_{t\in T}\sum_{u\in U}\sum_{w\in W}\sum_{d\in D}Q_{wdut}B_{wu}Tc_{ut}DS_{wd}+\sum_{t\in T}\sum_{u\in U}\sum_{a\in A}\sum_{k\in K}Q_{akut}B_{au}Tc_{ut}DS_{ak}. \tag{14}$$

Inventory holding costs:

$$\sum_{w\in W}\sum_{u\in U}\sum_{t\in T}R_{wut}WH_{wut}+\sum_{d\in D}\sum_{u\in U}\sum_{t\in T}R_{dut}DH_{dut}. \tag{15}$$

A.2. Constraints

All constraints of the proposed generic model are presented as follows. Clearly, this general model is a mixed integer linear programming (MILP) model:

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

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

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

$$\sum_{d\in D} Q_{dcut} B_{du} + \sum_{f\in F} Q_{fcut} B_{fu} + \sum_{w\in W} Q_{wcut} B_{wu} \geqslant 0.7 \times D_{cut}, \quad \forall t \in T,\ \forall u \in U,\ \forall c \in C, \tag{19}$$

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

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

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

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

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

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

$$Rc + Rm + Rr + Rp = 1, \tag{62}$$

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

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

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

Constraints (16)–(25), (62), (26)–(28) are balance constraints. Reviewing Fig. 2 reveals the necessity of balancing in each entity. At each node, all entering flows of every product per period should be equal to all issuing flows from that node. For all the entities in Fig. 2, this constraint should be seen. Hence, constraints (16) are the balance constraints of manufacturers, constraints (17)–(21) are for warehouses (17), distributors (18), customers considering 70% service level requirement (19), disassembly centers' inputs (20), and disassembly centers' outputs (21), respectively. Again, constraints (22)–(25), (62), (26)–(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 1 (constraint 62). It should be mentioned about constraint (62) that return products are collected from customers to disassembly centers, and then there are four options available based on the quality of return products: Some are sent to suppliers for recycling (Rc), some are sent to manufacturers for remanufacturing (Rm), some are appropriate to be sent to second markets and so they are transferred to redistributors after repairs (Rr), and finally, the rest are disposed through the disposal centers to ensure environmentally friendly disposal (Rp).

$$\sum_{f\in F} Q_{sfut} B_{su} \leqslant SC_{sut} L_s, \quad \forall t \in T,\ \forall u \in U,\ \forall s \in S, \tag{29}$$

$$\left(\sum_{d\in D} Q_{fdut} B_{fu} + \sum_{w\in W} Q_{fwut} B_{fu} + \sum_{c\in C} Q_{fcut} B_{fu} + \sum_{k\in K} Q_{fkut} B_{fu}\right) Fh_{fu} \leqslant FC_{fut} L_f, \quad \forall t \in T,\ \forall u \in U,\ \forall f \in F, \tag{30}$$

$$R_{wut} \leqslant SC_{wut} L_w, \quad \forall t \in T,\ \forall u \in U,\ \forall w \in W, \tag{31}$$

$$\sum_{f\in F} Q_{fdut} B_{fu} + \sum_{w\in W} Q_{wdut} B_{wu} + R_{du(t-1)} \leqslant DC_{dut} L_d, \quad \forall t \in T,\ \forall u \in U,\ \forall d \in D, \tag{32}$$

$$\sum_{s\in S} Q_{asut} B_{au} + \sum_{f\in F} Q_{afut} B_{au} + \sum_{r\in R} Q_{arut} B_{au} + \sum_{p\in P} Q_{aput} B_{au} \leqslant AC_{aut} \times L_a, \quad \forall t \in T,\ \forall u \in U,\ \forall a \in A, \tag{33}$$

$$\sum_{k\in K} Q_{rkut} B_{ru} \leqslant RC_{rut} \times L_r, \quad \forall t \in T,\ \forall u \in U,\ \forall r \in R, \tag{34}$$

$$\sum_{a\in A} Q_{asut} B_{au} \leqslant SRC_{sut} \times L_s, \quad \forall t \in T,\ \forall u \in U,\ \forall s \in S, \tag{35}$$

$$\sum_{a\in A} Q_{aput} B_{au} \leqslant PC_{put} \times L_p, \quad \forall t \in T,\ \forall u \in U,\ \forall p \in P, \tag{36}$$

$$\sum_{f\in F} Q_{fwut} B_{fu} \leqslant WC_{wut} \times L_w, \quad \forall t \in T,\ \forall u \in U,\ \forall w \in W. \tag{37}$$

Constraints (29)–(37) are capacity constraints controlling maximum flows that can enter into or issue from each entity (node). Constraint (29) controls suppliers' output capacities for each product per period. Constraints (30)–(37) are for capacities of manufacturers, warehouses, distributors, redistributors, suppliers, disposal centers, and warehouses inputs.

$$Li_{sf} \leqslant \sum_{u\in U}\sum_{t\in T} Q_{sfut} \leqslant M Li_{sf}, \quad \forall s \in S,\ \forall f \in F, \tag{38}$$

$$Li_{fd} \leqslant \sum_{u\in U}\sum_{t\in T} Q_{fdut} \leqslant M Li_{fd}, \quad \forall f \in F,\ \forall d \in D, \tag{39}$$

$$Li_{fw} \leqslant \sum_{u\in U}\sum_{t\in T} Q_{fwut} \leqslant M Li_{fw}, \quad \forall f \in F,\ \forall w \in W, \tag{40}$$

$$Li_{fc} \leqslant \sum_{u\in U}\sum_{t\in T} Q_{fcut} \leqslant M Li_{fc}, \quad \forall f \in F,\ \forall c \in C, \tag{41}$$

$$Li_{fk} \leqslant \sum_{u\in U}\sum_{t\in T} Q_{fkut} \leqslant M Li_{fk}, \quad \forall f \in F,\ \forall k \in K, \tag{42}$$

$$Li_{fr} \leqslant \sum_{u\in U}\sum_{t\in T} Q_{frut} \leqslant M Li_{fr}, \quad \forall r \in R,\ \forall f \in F, \tag{43}$$

$$Li_{wd} \leqslant \sum_{u\in U}\sum_{t\in T} Q_{wdut} \leqslant M Li_{wd}, \quad \forall w \in W,\ \forall d \in D, \tag{44}$$

$$Li_{wc} \leqslant \sum_{u\in U}\sum_{t\in T} Q_{wcut} \leqslant M Li_{wc}, \quad \forall w \in W,\ \forall c \in C, \tag{45}$$

$$Li_{wk} \leqslant \sum_{u\in U}\sum_{t\in T} Q_{wkut} \leqslant M Li_{wk}, \quad \forall w \in W,\ \forall k \in K, \tag{46}$$

$$Li_{wr} \leqslant \sum_{u\in U}\sum_{t\in T} Q_{wrut} \leqslant M Li_{wr}, \quad \forall w \in W,\ \forall r \in R, \tag{47}$$

$$Li_{dc} \leqslant \sum_{u\in U}\sum_{t\in T} Q_{dcut} \leqslant M Li_{dc}, \quad \forall d \in D,\ \forall c \in C, \tag{48}$$

$$Li_{ca} \leqslant \sum_{u\in U}\sum_{t\in T} Q_{caut} \leqslant M Li_{ca}, \quad \forall a \in A,\ \forall c \in C, \tag{49}$$

$$Li_{as} \leqslant \sum_{u\in U}\sum_{t\in T} Q_{asut} \leqslant M Li_{as}, \quad \forall s \in S,\ \forall a \in A, \tag{50}$$

$$Li_{af} \leqslant \sum_{u\in U}\sum_{t\in T} Q_{afut} \leqslant M Li_{af}, \quad \forall f \in F,\ \forall a \in A, \tag{51}$$

$$Li_{ar} \leqslant \sum_{u\in U}\sum_{t\in T} Q_{arut} \leqslant M Li_{ar}, \quad \forall r \in R,\ \forall a \in A, \tag{52}$$

$$Li_{ap} \leqslant \sum_{u\in U}\sum_{t\in T} Q_{aput} \leqslant M Li_{ap}, \quad \forall p \in P,\ \forall a \in A, \tag{53}$$

$$Li_{rk} \leqslant \sum_{u\in U}\sum_{t\in T} Q_{rkut} \leqslant M Li_{rk}, \quad \forall k \in K,\ \forall r \in R. \tag{54}$$

Constraints (38)–(54) manage the links between entities. For instance, when the left of constraint (38) is considered and when there are no flows between a supplier and a manufacturer (all products in all periods), there should be no link between both entities. When, based on the right-hand-side of the same constraint, there is no real link/shipping way between the supplier and the manufacturer, flows cannot be expected there. Hence, such constraints guarantee no links between nodes without actual real flows and no flows between two nodes sans an actual link.

$$\sum_{s\in S} L_s \leqslant S, \tag{55}$$

$$\sum_{f\in F} L_f \leqslant F, \tag{56}$$

$$\sum_{d\in D} L_d \leqslant D, \tag{57}$$

$$\sum_{w\in W} L_w \leqslant W, \tag{58}$$

$$\sum_{a\in A} L_a \leqslant A, \tag{59}$$

$$\sum_{r\in R} L_r \leqslant R, \tag{60}$$

$$\sum_{p\in P} L_p \leqslant P. \tag{61}$$

Constraints (55)–(61) manage the maximum number of allowable locations. Though there are limitations on the number of activated locations, they cope with them and do not allow a supply chain to establish more nodes than relative possible limitations.

References

[1] H. Winkler, Closed-loop production systems—a sustainable supply chain approach, CIRP J. Manuf. Sci. Technol. 4 (3) (2011) 243–246.
[2] B.M. Beamon, Designing the green supply chain, Logistics Inf. Manage. 12 (4) (1999) 332–342.
[3] S. Chopra, P. Meindl, Supply Chain Management: Strategy, Planning and Operation, third ed. book., Pearson Prentice Hall Inc, 2007. ISBN 81-7758-003-5.
[4] J. Krarup, P.M. Pruzan, The simple plant location problem: survey and synthesis, Eur. J. Oper. Res. 12 (1) (1983) 36–81.
[5] A. Schrijver, Combinatorial Optimization: Polyhedra and Efficiency, Springer, Berlin, Germany, 2003.
[6] M.I. Gomes Salema, A.P. Barbosa-Povoa, A.Q. Novais, An integrated model for the design and planning of supply chains with product return, Comput. Aided Chem. Eng. 21 (2006) 2129–2134.
[7] M.I. Gomes Salema, A.P. Barbosa-Povoa, A.Q. Novais, An optimization model for the design of a capacitated multi-product reverse logistics network with uncertainty, Eur. J. Oper. Res. 179 (3) (2007) 1063–1077.
[8] A.C.S. Amaro, A.P. Barbosa-Póvoa, Optimal planning of closed loop supply chains: a discrete versus a continuous-time formulation, Comput. Aided Chem. Eng. 24 (1) (2007) 673–678.
[9] A.C.S. Amaro, A.P. Barbosa-Póvoa, Close loop supply chains: managing product recovery portfolio, Comput. Aided Chem. Eng. 21 (2006) 1875–1880.
[10] R.D. Kusumastuti, R. Piplani, G.H. Lim, Redesigning closed-loop service network at a computer manufacturer: a case study, Int. J. Prod. Econ. 111 (2) (2008) 244–260.
[11] Y. Zhou, S. Wang, Generic model of reverse logistics network design, J. Transp. Syst. Eng. Inf. Technol. 8 (3) (2008) 71–78.

[12] A.C.S. Amaro, A.F.P.D. Barbosa-Póvoa, The effect of uncertainty on the optimal closed-loop supply chain planning under different partnerships structure, Comput. Chem. Eng. 33 (12) (2009) 2144–2158.
[13] A. Mutha, S. Pokharel, Strategic network design for reverse logistics and remanufacturing using new and old product modules, Comput. Ind. Eng. 56 (1) (2009) 334–346.
[14] M.I. Gomes Salema, A.P. Barbosa-Povoa, A.Q. Novais, Simultaneous design and planning of supply chains with reverse flows: a generic modelling framework, Eur. J. Oper. Res. 203 (2) (2010) 336–349.
[15] M.S. Pishvaee, F. Jolai, J. Razmi, A stochastic optimization model for integrated forward/reverse logistics network design, J. Manuf. Syst. 28 (4) (2009) 107–114.
[16] X. Li, T.E. Marlin, Robust supply chain performance via model predictive control, Comput. Chem. Eng. 33 (12) (2009) 2134–2143.
[17] M. El-Sayed, N. Afia, A. El-Kharbotly, A stochastic model for forward–reverse logistics network design under risk, Comput. Ind. Eng. 58 (3) (2010) 423–431.
[18] P. Georgiadis, An integrated system dynamics model for strategic capacity planning in closed-loop recycling networks: a dynamic analysis for the paper industry, Simul. Model. Pract. Theory 32 (2013) 116–137.
[19] M. Besiou, P. Georgiadis, L.N. Van Wassenhove, Official recycling and scavengers: Symbiotic or conflicting?, Eur J. Oper. Res. 218 (2) (2012) 563–576.
[20] P. Georgiadis, E. Athanasiou, The impact of two-product joint lifecycles on capacity planning of remanufacturing networks, Eur. J. Oper. Res. 202 (2) (2010) 420–433.
[21] P. Georgiadis, D. Vlachos, G. Tagaras, The impact of product lifecycle on capacity planning of closed-loop supply chains with remanufacturing, Prod. Oper. Manage. 15 (4) (2006) 514–527.
[22] P. Georgiadis, E. Athanasiou, Flexible long-term capacity planning in closed-loop supply chains with remanufacturing, Eur. J. Oper. Res. 225 (1) (2013) 44–58.
[23] G. Kannan, A. Noorul Haq, M. Devika, Analysis of closed loop supply chain using genetic algorithm and particle swarm optimization, Int. J. Prod. Res. 47 (5) (2009) 1175–1200.
[24] G. Kannan, P. Sasikumar, K. Devika, A genetic algorithm approach for solving a closed loop supply chain model: a case of battery recycling, Appl. Math. Model. 34 (2010) 655–670.
[25] H.F. Wang, H.W. Hsu, A closed-loop logistic model with a spanning tree based genetic algorithm, Comput. Oper. Res. 37 (2) (2010) 376–389.
[26] F. Dehghanian, S. Mansour, Designing sustainable recovery network of end-of-life products using genetic algorithm, Resour. Conserv. Recycl. 53 (10) (2009) 559–570.
[27] M.S. Pishvaee, R. Zanjirani Farahani, W. Dullaert, A memetic algorithm for bi-objective integrated forward/reverse logistics network design, Comput. Oper. Res. 37 (6) (2010) 1100–1112.
[28] D.H. Lee, M. Dong, A heuristic approach to logistics network design for end-of-lease computer products recovery, Transp. Res. E Logistics Transp. Rev. 44 (3) (2008) 455–474.
[29] V. Özkir, H. Basligil, Multi-objective optimization of closed-loop supply chains in uncertain environment, J. Cleaner Prod. 41 (2013) 114–125.
[30] E. Özceylan, T. Paksoy, A mixed integer programming model for a closed-loop supply chain network, Int. J. Prod. Res. 51 (3) (2013) 718–734.
[31] M. Ramezani, M. Bashiri, R. Tavakkoli-Moghaddam, A new multi-objective stochastic model for a forward/reverse logistic network design with responsiveness and quality level, Appl. Math. Model. 37 (1–2) (2013) 328–344.
[32] S.H. Amin, G. Zhang, An integrated model for closed-loop supply chain configuration and supplier selection: multi-objective approach, Expert Syst. Appl. 39 (8) (2012) 6782–6791.
[33] J. Wang, J. Zhao, X. Wang, Optimum policy in hybrid manufacturing/remanufacturing system, Comput. Ind. Eng. 60 (3) (2011) 411–419.
[34] T. Paksoy, T. Bektas, E. Özceylan, Operational and environmental performance measures in a multi-product closed-loop supply chain, Transp. Res. E 47 (4) (2011) 532–546.
[35] Q. Qiang, K. Ke, T. Anderson, J. Dong, The closed-loop supply chain network with competition, distribution channel investment, and uncertainties, Omega 41 (2) (2013) 186–194.
[36] L.T. Khajavi, S.M. Seyed-Hosseini, A. Makui, An integrated forward/reverse logistics network optimization model for multi-stage capacitated supply chain, iBusiness 3 (2) (2011) 229–235.
[37] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: 1995 IEEE International Conference on Neural Networks, Proceedings, vol. 4, 1995, pp. 1942–1948.
[38] J.H. Holland, Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence, second ed., MIT press, 1992.
[39] R.E. Hartl, A Global Convergence Proof for a Class of Genetic Algorithms, University of Technology, Vienna, 1990.
[40] N.J. Radcliffe, Genetic set recombination, Found. Genet. Algorithms 2 (1993) 203–219.
[41] J.C. Bean, Genetics and random keys for sequencing and optimization, ORSA J. Comput. 6 (2) (1994) 154–160.
[42] E. Özceylan, T. Paksoy, Fuzzy multi objective linear programming approach for optimizing a closed-loop supply chain network, Int. J. Prod. Res. 51 (8) (2013) 2443–2461.
[43] Y. Wang, T. Lu, C. Zhang, Integrated Logistics Network Design in Hybrid Manufacturing/Remanufacturing System Under Low-Carbon Restriction, Springer Berlin Heidelberg, 2013. In LISS, 111–121.
[44] H. Metta, F. Badurdeen, Optimized closed-loop supply chain configuration selection for sustainable product designs, in: IEEE Conference on Automation Science and Engineering (CASE), 2011, pp. 438–443.
[45] H. Soleimani, M. Seyyed-Esfahani, M.A. Shirazi, Designing and planning a multi-echelon multi-period multi-product closed-loop supply chain utilizing genetic algorithm, Int. J. Adv. Manuf. Technol. 68 (1–4) (2013) 917–931.
[46] X. Zhou, Z. Zhao, K. Zhou, Remanufacturing closed-loop supply chain network design based on genetic particle swarm optimization algorithm, J. Central South Univ. Technol. 19 (2012) 482–487.
[47] S.H. Amin, G. Zhang, A multi objective facility location model for closed-loop supply chain network under uncertain demand and return, Appl. Math. Model. 37 (2013) 4165–4167.
[48] T.M. Choi, Y. Li, L. Xu, Channel leadership, performance and coordination in closed-loop supply chains, Int. J. Prod. Econ. 146 (2013) 371–380.
[49] B.L. Miller, D.E. Goldberg, Genetic algorithms, tournament selection, and the effects of noise, Evol. Comput. 4 (2) (1996) 113–131.
[50] M.D. Vose, Generalizing the notion of schema in genetic algorithms, Artif. Intell. 50 (1) (1991) 385–396.
[51] J.R. Koza, Genetic programming: II: Automatic discovery of reusable programs, Artif. Life 1 (4) (1994) 439–441.
[52] C.R. Reeves, Modern Heuristic Techniques for Combinatorial Problems, Blackwell Scientific Publications, Oxford, 1993.
[53] M. Mitchell, An introduction to genetic algorithms, Cambridge, Massachusetts London, England, Fifth printing, vol. 3, 1999.
[54] T. Baeck, D.B. Fogel, Z. Michalewicz, Handbook of Evolutionary Computation, IOP Publishing Ltd., New York, 1997.
[55] W. Guang-Min, W. Zhong-Ping, W. Xian-Jia, C. Ya-Lin, Genetic algorithms for solving linear bilevel programming, in: Sixth International Conference on Parallel and Distributed Computing, Applications and Technologies, PDCAT 2005, IEEE, 2005, pp. 920–924.
[56] S. Du, W. Li, K. Cao, A learning algorithm of artificial neural network based on GA–PSO, The Sixth World Congress on Intelligent Control and Automation, WCICA 2006, vol. 1, IEEE, 2006, pp. 3633–3637.
[57] R.J. Kuo, Y.S. Han, A hybrid of genetic algorithm and particle swarm optimization for solving bi-level linear programming problem – a case study on supply chain model, Appl. Math. Model. 35 (8) (2011) 3905–3917.